

# **Efektivní algoritmy pro simulaci DE systémů**

## **Effective algorithms for simulation of Discrete-Event systems**

## Zadání diplomové práce

Student: **Bc. Simona Domesová**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 1103T031 Výpočetní matematika

Téma: **Efektivní algoritmy pro simulaci DE systémů**  
**Effective algorithms for simulation of Discrete-Event systems**

### Zásady pro vypracování:

Většina komplexních systémů reálného světa může být modelována jako DE (Discrete-Event) systémy. Příkladem mohou být např. dopravní systémy, počítačově-komunikační systémy, zásobovací systémy, výrobní linky, koherentní systémy, či PERT síť (Program Evaluation and Review Technique). Chování těchto systémů je identifikováno prostřednictvím posloupnosti diskretních událostí, které způsobují stavové změny.

### Postup práce:

1. Způsoby počítačové simulace DE systémů.
2. Statistická analýza DE systémů.
3. Přehled metod pro redukci rozptylu, s ohledem na kvantifikaci řídkých jevů.
4. Analýza citlivosti DE systémů.
5. PC implementace, případně porovnání s dalšími dostupnými algoritmy.

### Seznam doporučené odborné literatury:

R.Y. Rubinstein and D.P. Kroese. Simulation and the Monte Carlo Method, Wiley, 2007.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **prof. Ing. Radim Briš, CSc.**

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015



doc. RNDr. Jiří Bouchala, Ph.D.  
*vedoucí katedry*



prof. RNDr. Václav Snášel, CSc.  
*děkan fakulty*

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

V Ostravě 7. května 2015

.....

Ráda bych touto cestou poděkovala vedoucímu diplomové práce prof. Ing. Radimu Brišovi, CSc. za věnovaný čas, cenné rady a poskytnutí podkladů pro vypracování této práce.

## Abstrakt

Tato diplomová práce se zabývá simulací DE systémů, tedy takových systémů, k jejichž změnám dochází v diskrétních časových okamžicích. Simulace je založena na metodě Monte Carlo a její efektivita je zajištěna použitím metod redukce rozptylu a pokročilými způsoby implementace. Použitými metodami redukce rozptylu jsou metoda antitických náhodných veličin, metoda řídicích veličin, podmíněná metoda Monte Carlo a metoda *importance sampling*, je kladen důraz na kvantifikaci řídkých jevů. Práce se dále zabývá analýzou citlivosti DE systémů pomocí metody centrálních diferencí a metody *score function*. Všechny zmíněné metody byly implementovány v jazyce Matlab a poté paralelizovány na GPU pomocí technologie CUDA. Simulační metody byly dále aplikovány na problémy z oblasti spolehlivosti systémů.

**Klíčová slova:** analýza citlivosti, DE systém, GPU paralelizace, Monte Carlo, redukce rozptylu, spolehlivost systému, stochastická simulace, řídký jev

## Abstract

This master thesis focuses on DE systems simulation, the changes of these systems occur at discrete points of time. The simulation is based on the Monte Carlo method and its effectivity is ensured by using variance reduction techniques and advanced methods of implementation. The variance reduction methods include antithetic random variables, control variables, conditional Monte Carlo and importance sampling, the emphasis is on rare event probabilities estimation. The thesis also deals with sensitivity analysis of DE systems using central differences and the score function method. All of the methods were implemented in Matlab and GPU parallelized using CUDA technology. The simulation methods were also applied to problems of system reliability.

**Keywords:** sensitivity analysis, DE system, GPU parallelization, Monte Carlo, variance reduction, system reliability, stochastic simulation, rare event

## Seznam použitých zkratk a symbolů

CE	– Cross-Entropy
CRN	– Common Random Numbers
CUDA	– Compute Unified Device Architecture
DE	– Discrete-Event
DEDS	– Discrete-Event Dynamic System
DESS	– Discrete-Event Static System
IS	– Importance Sampling
LR	– Likelihood Ratio
MC	– Monte Carlo
NEF	– Natural Exponential Family
NV	– Náhodná veličina
PERT	– Program Evaluation and Review Technique
PMC	– Permutation Monte Carlo
RSD	– Relative Standard Deviation
RE	– Relative Error
SPB	– Single Parameter Biasing
VM	– Variance Minimization (minimalizace rozptylu)
Ber	– Bernoulliho rozdělení
Cov	– Kovariance
$\mathbb{E}$	– Střední hodnota
Exp	– Exponenciální rozdělení
$I_A$	– NV rovna 1, nastane-li jev $A$ , a 0 v opačném případě
$\mathcal{N}$	– Normální rozdělení
$\mathbb{P}$	– Pravděpodobnost
$\mathcal{U}$	– Rovnoměrné rozdělení
Var	– Rozptyl
Weib	– Weibullovo rozdělení

## Obsah

<b>1</b>	<b>Úvod</b>	<b>5</b>
<b>2</b>	<b>DE systémy a jejich simulace</b>	<b>6</b>
2.1	Modely systémů a jejich klasifikace . . . . .	6
2.2	Stav systému . . . . .	6
2.3	Simulace DE systémů . . . . .	7
2.4	Výkonnostní funkce . . . . .	10
<b>3</b>	<b>Statistická analýza DE systémů</b>	<b>12</b>
3.1	Výsledek simulace . . . . .	12
3.2	Metoda Monte Carlo . . . . .	12
3.3	Řídké jevy . . . . .	19
<b>4</b>	<b>Přehled metod pro redukci rozptylu</b>	<b>20</b>
4.1	Metoda antitických náhodných veličin . . . . .	20
4.2	Metoda řídicích veličin . . . . .	24
4.3	Podmíněná metoda Monte Carlo . . . . .	29
4.4	Importance sampling . . . . .	31
<b>5</b>	<b>Analýza citlivosti DE systémů</b>	<b>42</b>
5.1	Metoda centrálních diferencí . . . . .	42
5.2	Score Function . . . . .	45
<b>6</b>	<b>Implementace a paralelizace</b>	<b>50</b>
6.1	Paralelizace výkonnostní funkce pomocí CUDA . . . . .	51
<b>7</b>	<b>Aplikace I - dynamický systém se dvěma procesními proměnnými</b>	<b>53</b>
7.1	Specifikace systému . . . . .	53
7.2	Simulace systému . . . . .	55
7.3	Implementace a paralelizace . . . . .	58
7.4	Statistická analýza výsledků simulace . . . . .	59
7.5	Redukce rozptylu . . . . .	61
<b>8</b>	<b>Aplikace II - spolehlivost systémů s nezávislými prvky</b>	<b>68</b>
8.1	Specifikace systému . . . . .	68
8.2	Dynamická formulace . . . . .	70
8.3	Použité metody redukce rozptylu . . . . .	70
8.4	Implementace a paralelizace . . . . .	73
8.5	Test použitých algoritmů . . . . .	75
8.6	Analýza citlivosti . . . . .	81
<b>9</b>	<b>Závěr</b>	<b>84</b>

---

<b>10 Literatura</b>	<b>86</b>
<b>Přílohy</b>	<b>88</b>
<b>A Přehled používaných rozdělení pravděpodobnosti</b>	<b>88</b>
A.1 Rovnoměrné rozdělení . . . . .	88
A.2 Bernoulliho rozdělení . . . . .	88
A.3 Exponenciální rozdělení . . . . .	88
A.4 Weibullovo rozdělení . . . . .	88
A.5 Normální rozdělení . . . . .	88
<b>B Výsledky testování a vybrané zdrojové kódy</b>	<b>89</b>
B.1 Výsledky analýzy citlivosti . . . . .	89
B.2 Vybrané zdrojové kódy . . . . .	90
B.3 Test algoritmů používaných v sekci 8.5 . . . . .	92
<b>C Příloha na CD</b>	<b>95</b>



## Seznam tabulek

3.1	Tabulka projektu . . . . .	15
3.2	Výsledky příkladu 3.4 získané metodou MC . . . . .	17
4.1	Výsledky příkladu 4.5 pro $N = 10^6, \alpha = 0.05, \varepsilon = 0.01$ . . . . .	23
4.2	Výsledky příkladu 4.5 pro $N = 10^6, \alpha = 0.05, \varepsilon = 0.01$ . . . . .	27
4.3	Výsledky příkladu 4.15 pro $N = 10^6, \alpha = 0.05, \varepsilon = 0.01$ . . . . .	31
4.4	Výsledky příkladu 4.25 pro $N = 10^6, \alpha = 0.05, \varepsilon = 0.01$ . . . . .	39
4.5	Porovnání výsledků metod MC a IS . . . . .	41
7.1	Funkce $\Theta(t)$ v závislosti na konfiguraci systému . . . . .	55
7.2	Parametry úlohy . . . . .	55
7.3	Porovnání výpočetních časů, uvedené časy jsou v sekundách . . . . .	59
7.4	Odhady hodnot $F_v(T), F_p(T)$ a $F_t(T)$ pomocí metody MC . . . . .	60
7.5	Odhady hodnot $F_v(T), F_p(T)$ a $F_t(T)$ metodou antitetických NV . . . . .	62
7.6	Závislost $Q_a$ na $a_1, a_2, a_3$ . . . . .	62
7.7	Odhady hodnot $F_v(T), F_p(T)$ a $F_t(T)$ s použitím řídící veličiny $C$ . . . . .	63
7.8	Odhady hodnot $F_v(T), F_p(T)$ a $F_t(T)$ s použitím veličin $C_v, C_p$ a $C_t$ . . . . .	64
7.9	Odhady hodnot $F_v(T), F_p(T)$ a $F_t(T)$ metodou IS pro $\mathcal{N}(\cdot, 1)$ . . . . .	65
7.10	Odhady hodnot $F_v(T)$ a $F_p(T)$ metodou IS pro $\text{Ber}(u)$ . . . . .	66
7.11	Porovnání metod redukce rozptylu . . . . .	67
8.1	Výpočetní čas jednotlivých implementací funkce $G(\pi)$ . . . . .	73
8.2	Výsledky příkladu 8.6 . . . . .	77
8.3	Výsledky příkladu 8.6 řešeného metodou MC . . . . .	78
8.4	Zadání příkladu 8.8 . . . . .	80
8.5	Výsledky příkladu 8.8 . . . . .	80
B.1	Odhad $\nabla \ell(\theta)$ pomocí metody centrálních diferencí . . . . .	89
B.2	Odhad $\nabla \ell_2(\mu)$ pomocí metody centrálních diferencí . . . . .	89
B.3	Odhad $\nabla \ell(\theta)$ pomocí metody SF . . . . .	90
B.4	Odhad $\nabla \ell_2(\mu)$ pomocí metody SF . . . . .	90
B.5	Algoritmus 1 - metoda IS pro statickou formulaci (příklad 8.6) . . . . .	92
B.6	Algoritmus 2 - metoda IS pro dynamickou formulaci (příklad 8.6) . . . . .	92
B.7	Algoritmus 3a - sekvenční implementace metody PMC (příklad 8.6) . . . . .	93
B.8	Algoritmus 3b - CUDA implementace metody PMC (příklad 8.6) . . . . .	93
B.9	Algoritmus 3c - hybridní implementace metody PMC (příklad 8.6) . . . . .	93
B.10	Výsledky příkladu 8.7 . . . . .	94

## Seznam obrázků

2.1	Schéma systému hromadné obsluhy . . . . .	8
2.2	Vztahy mezi jednotlivými událostmi . . . . .	9
2.3	Vývoj počtu osob ve frontách . . . . .	9
2.4	Schéma systému z příkladu 2.3 . . . . .	10
2.5	Schéma systému z příkladu 2.4 . . . . .	11
3.1	PERT síť odpovídající tabulce 3.1 . . . . .	15
3.2	Vizualizace výsledků simulace . . . . .	16
3.3	Vývoj počtu osob ve frontách . . . . .	18
3.4	Schéma systému z příkladu 3.6 se znázorněním parametrů $\lambda_i$ . . . . .	18
4.1	Závislost náhodných veličin $Y$ a $Y^*$ , náhodný výběr o rozsahu $10^4$ . . . . .	23
4.2	Závislost náhodných veličin $Y$ a $C$ , náhodný výběr o rozsahu $10^4$ . . . . .	28
4.3	Histogramy odhadů hodnoty $\ell$ získaných z $10^3$ simulací . . . . .	28
5.1	Ilustrace vlivu jednotlivých komponent na hodnotu $\ell$ . . . . .	45
5.2	Odhad průběhu funkce $(\nabla \ell(\theta))_6$ pomocí metody SF . . . . .	48
5.3	Odhad průběhu funkce $\ell(\theta)$ pomocí metody IS . . . . .	49
7.1	Schéma systému . . . . .	53
7.2	Příklad simulace . . . . .	58
7.3	Závislost výpočetního času na velikosti dávky . . . . .	59
7.4	Meze spolehlivosti odhadu stanovené metodou MC . . . . .	60
7.5	Odhadovaný průběh funkcí $F_v(t)$ , $F_p(t)$ a $F_t(t)$ pro $t \in \langle 0, T \rangle$ ; $N = 10^6$ . . . . .	61
7.6	Meze spolehlivosti odhadu stanovené metodou antitických NV . . . . .	63
7.7	Meze spolehlivosti odhadu stanovené metodou řídicích veličin . . . . .	64
7.8	Meze spolehlivosti odhadu stanovené metodou IS . . . . .	66
8.1	Znázornění systému z příkladu 8.6 . . . . .	76
8.2	Porovnání efektivity jednotlivých algoritmů . . . . .	77
8.3	Metoda SPB . . . . .	78
8.4	Znázornění systému z příkladu 8.7 . . . . .	79
8.5	Porovnání efektivity algoritmů 1 a 3b . . . . .	79
8.6	Znázornění systému z příkladu 8.8 . . . . .	80
8.7	Analýza citlivosti systému z příkladu 8.6 . . . . .	82
8.8	Analýza citlivosti systému z příkladu 8.7 . . . . .	83

## 1 Úvod

S DE systémy (z anglického *discrete-event systems*) se setkáváme v mnoha oblastech běžného života i inženýrské praxe. Mezi typické DE systémy patří systémy hromadné obsluhy, komunikační sítě, výrobní linky, zásobovací systémy a mnohé další. Jmenované systémy jsou složeny ze samostatných prvků a k jejich změnám dochází v diskrétních časových okamžicích. Stochastická počítačová simulace těchto systémů má v praxi široké uplatnění, neboť na rozdíl od reálného testování systémů ji lze provést s minimálními časovými a finančními nároky.

Mezi hlavní cíle této práce patří využít při simulaci DE systémů metody redukce rozptylu založené na metodě Monte Carlo. Jedná se o skupinu simulačních metod, které mohou být aplikovány na široké spektrum stochastických i deterministických úloh za účelem zvýšení efektivity simulace. Dalším úkolem práce je vybrat vhodné metody analýzy citlivosti a použít je pro vyhodnocení vlivu jednotlivých parametrů systému na chování systému jako celku.

Druhá kapitola popisuje, jak simulovat chování DE systémů, a vysvětluje související pojmy. Po provedení simulace je třeba vyhodnotit simulační data a získat tak o systému co nejvíce informací. Statistickým vyhodnocením simulačních dat pomocí metody Monte Carlo se zabývá kapitola 3. Kapitoly 4 a 5 se věnují metodám redukce rozptylu a analýzy citlivosti. V případě všech metod je uvedena veškerá teorie potřebná k jejich pochopení a použití metod je průběžně ilustrováno na jednoduchých příkladech.

Kapitola 6 hovoří obecně o způsobech implementace simulačních metod a klade si za cíl využít potenciál rychle se rozvíjejících výpočetních možností. Studované způsoby simulace a implementace jsou dále aplikovány na komplexní problémy. Kapitola 7 se zabývá simulací dynamického systému se dvěma procesními proměnnými, který byl v řadě odborných článků použit pro testování efektivity různých způsobů simulace. V kapitole 8 je řešena spolehlivost systémů s nezávislými prvky, tato problematika má široké uplatnění ve všech oblastech, v nichž je třeba zajistit téměř bezchybný chod systémů.

Přehled vybraných rozdělení pravděpodobnosti včetně používaného značení je uveden v příloze.

## 2 DE systémy a jejich simulace

Cílem této kapitoly je vysvětlit, co je to DE systém (z anglického *discrete-event system*) a jak lze simulovat jeho chování. Nejprve je však třeba objasnit, co znamená samotný pojem systém. Obecně se jedná o soubor komponent, které dohromady tvoří jeden funkční celek. V praxi se může jednat například o zásobovací systém, výrobní linku či počítačovou síť.

Podkladem pro tuto kapitolu jsou zejména publikace [1] a [2].

### 2.1 Modely systémů a jejich klasifikace

Před studiem reálného systému je třeba nejprve vytvořit model, který tento systém reprezentuje. Pomocí modelu je dále řešen problém týkající se původního systému.

Při vytváření modelu je kladen důraz na jeho jednoduchost a srozumitelnost, zároveň je však nutné, aby model co nejlépe vypovídal o realitě. Tyto dva požadavky jsou protichůdné, proto je třeba zvolit vhodný kompromis. Obvykle jsou zachovány hlavní vlastnosti systému, zejména ty, které jsou důležité pro řešení problému, a zanedbány ty, jež nemají velký vliv na chování systému a zapříčiňují nepřehlednost modelu.

Je-li zvolen vhodný model, lze přejít k řešení problému. V některých případech je možné získat řešení analyticky, mnohdy je však vhodnější přejít k numerickému řešení. Tato práce se zabývá řešením úloh pomocí stochastické počítačové simulace, která spočívá v opakované realizaci možného chování modelu a následném statistickém vyhodnocení těchto realizací.

Systémy lze primárně dělit na statické a dynamické, dynamické systémy se na rozdíl od statických vyvíjejí v čase. Dynamické systémy lze dále dělit na spojité a diskrétní. Zatímco spojitý dynamický systém se může vyvíjet spojitě v čase, ke změnám stavu diskrétního dynamického systému dochází pouze v diskrétních časových okamžicích. Typickými příklady diskrétních dynamických systémů jsou systémy hromadné obsluhy, stav systému může určovat například počet osob v jednotlivých frontách a diskrétní změnou stavu systému může být příchod zákazníka či jeho obsloužení, viz příklad 2.2.

V této práci budou simulovány výhradně diskrétní statické systémy (zkráceně DESS z anglického *discrete-event static system*) a diskrétní dynamické systémy (zkráceně DEDS z anglického *discrete-event dynamic system*). Dále tedy bude systémem myšlen DE systém a budou rozlišovány pouze statické a dynamické systémy. Rovněž nebude kladen důraz na zjednodušení způsobená volbou modelu oproti původními reálnému systému, pojmy „systém“ a „model“ proto budou ztotožňovány.

### 2.2 Stav systému

Statickému systému o  $n$  komponentách je možné přiřadit stavový vektor  $(x_1, \dots, x_n)$ , jehož prvky reprezentují stavy jednotlivých komponent. Množina možných stavů komponenty může být obecně spojitá, často se však jedná pouze o množinu diskrétních hodnot. V mnoha aplikacích je množinou možných stavů komponenty dvouprvková množina

$\{0, 1\}$ , přičemž hodnota 1 symbolizuje, že je tato komponenta v provozu, a 0 značí její poruchu.

Dynamický systém se vyvíjí v čase, stavový vektor dynamického systému se tedy rovněž v čase mění. Konkrétnímu času  $t$  je tak přiřazen stavový vektor  $(x_1(t), \dots, x_n(t))$ , jehož prvky reprezentují stavy jednotlivých komponent v čase  $t$ .

V případě stochastických simulací je stav DESS chápán jako náhodný vektor  $\mathbf{X} = (X_1, \dots, X_n)$ . Vektor  $(x_1, \dots, x_n)$  je v tomto kontextu konkrétní realizací náhodného vektoru  $\mathbf{X}$ . Množina všech možných stavů systému se nazývá stavový prostor a bude dále značena symbolem  $\Omega$ .

Chování dynamických systémů může být chápáno jako náhodný proces  $\{\mathbf{X}_t, t \in \mathcal{T}\}$ , kde  $\mathbf{X}_t = ((X_1)_t, \dots, (X_n)_t)$  a  $\mathcal{T}$  je indexová množina, která je v případě DEDS diskrétní. Konkrétním pozorováním náhodného procesu  $\mathbf{X}_t$  je posloupnost  $\{x_t, t \in \mathcal{T}\}$ .

Základy teorie pravděpodobnosti a přehledně zpracovanou teorii týkající se náhodných veličin a procesů lze nalézt v [3].

## 2.3 Simulace DE systémů

Model statického systému určuje rozdělení pravděpodobnosti stavového vektoru  $\mathbf{X}$ . Stochastická simulace DESS tedy spočívá ve vygenerování konkrétní realizace  $(x_1, \dots, x_n) \in \Omega$  z rozdělení náhodného vektoru  $\mathbf{X}$ .

Ke změnám stavu DEDS dochází pouze ve spočetném množství časových okamžiků. Výsledkem simulace je tak posloupnost  $\{x_t, t \in \mathcal{T}\}$ , neboli konkrétní realizace náhodného procesu  $\mathbf{X}_t$ . V případě systémů, které pro  $t \rightarrow \infty$  vykazují určitý druh ustáleného chování, je možné vykonat jeden dlouhý běh simulace a následně přejít k vyhodnocení výsledků například pomocí tzv. regenerativní metody, viz [1]. V této práci budou dále studovány obecné systémy a bude prováděna tzv. simulace v konečném časovém horizontu, která spočívá v mnohonásobném opakování simulace chodu systému v intervalu  $t \in \langle 0, T \rangle$ , kde  $T \in \mathbb{R}^+$ , a následném vyhodnocení získaných dat.

### 2.3.1 Event clock

Vhodným mechanismem pro simulaci DEDS je tzv. *event clock*. Tento způsob simulace předpokládá, že je chování modelu řízeno několika typy opakujících se událostí, přičemž každá událost může vyvolat naplánování dalších událostí. Všechny nově naplánované události jsou přidány do seznamu událostí. Jednotlivé řádky tohoto seznamu obsahují informaci o tom, o jaký typ události se jedná, v jakém čase k ní dojde, a případně i další parametry události. Simulace pomocí *event clock* se řídí algoritmem 2.1, který je třeba chápat pouze jako obecný princip simulace, při aplikaci na konkrétní systém může být vhodně algoritmus pozměnit.

---

#### Algoritmus 2.1 (Event clock).

1. Inicializovat proměnnou  $t = 0$ . Vytvořit počáteční seznam událostí.
2. Zaznamenat stav systému v čase  $t$ , tedy hodnotu  $x(t)$ .

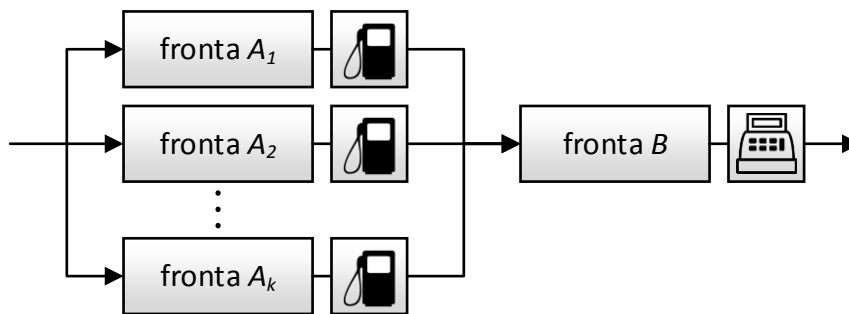
3. Seřadit seznam událostí podle času naplánovaných událostí.
4. Provést první událost na seznamu, nastavit proměnnou  $t$  na čas této události.
5. Aktuální událost vyvolá  $m$  dalších událostí,  $m \geq 0$ . Pro každou z plánovaných událostí
  - (a) určit její čas  $t_i > t$  a její typ,
  - (b) přidat tuto událost do seznamu událostí.
6. Smazat aktuální událost ze seznamu a pokračovat krokem 2.

Simulaci lze ukončit například tehdy, když čas první události na seznamu překročí předem stanovenou délku simulace. Simulace je rovněž ukončena, pokud již na seznamu nejsou žádné události.

Zbývá poznamenat, že 4. bod algoritmu 2.1 může zahrnovat například změnu času některých událostí ze seznamu, nebo jejich úplné smazání, nemusí tedy dojít ke každé události, která byla naplánována.

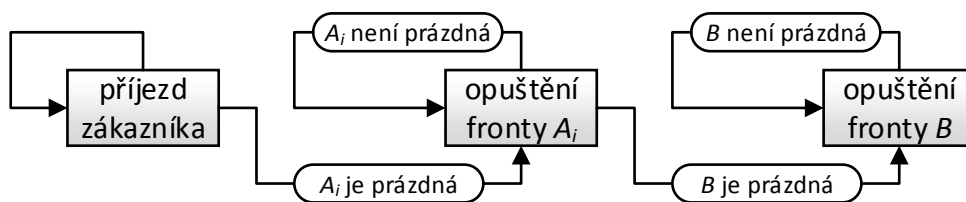
Následující příklad se věnuje simulaci systému hromadné obsluhy pomocí mechanismu *event clock*. Jedná se o soustavu front, která může znázorňovat například pohyb zákazníků na čerpací stanici.

**Příklad 2.2** (Systém hromadné obsluhy, část 1). Schéma systému znázorňuje obrázek 2.1. Příjezd zákazníků se řídí Poissonovým procesem, nový zákazník se zařadí vždy do nejkratší z front  $A_1, \dots, A_k$ . Když se zákazník dostane na řadu, načerpá pohonné hmoty a poté se přesune k pokladně, tedy zařadí se do fronty  $B$ . Doba čerpání pohonných hmot a doba obsluhy u pokladny jsou modelovány jako náhodné veličiny pocházející z Weibullova rozdělení.



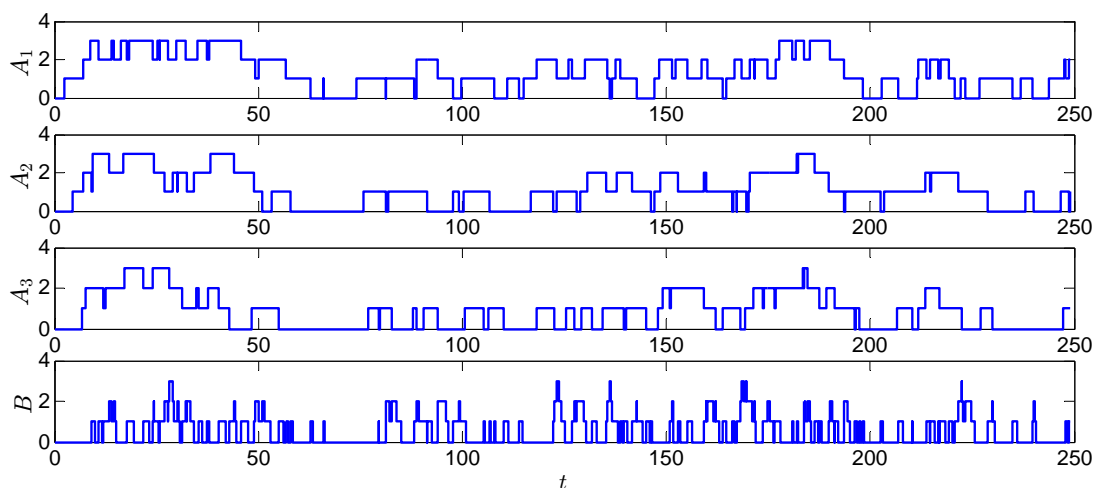
Obrázek 2.1: Schéma systému hromadné obsluhy

Nastávají tři typy událostí, příjezd zákazníka, opuštění některé z front  $A_1, \dots, A_k$  a opuštění fronty  $B$ . Vzájemné vztahy mezi jednotlivými událostmi znázorňuje schéma na obrázku 2.2. Šipka vedoucí z události  $X$  do události  $Y$  značí, že událost  $X$  vyvolá za uvedené podmínky naplánování události  $Y$ .



Obrázek 2.2: Vztahy mezi jednotlivými událostmi

Simulace byla provedena pro  $k = 3$ , doba do příjezdu zákazníka pochází z  $\text{Exp}(0.5)$ , doba čerpání pohonných hmot z  $\text{Weib}(0.2, 2)$  a doba obsluhy u pokladny z  $\text{Weib}(1, 2)$ . Časovou jednotkou je minuta, průměrná doba mezi příchodem zákazníků tedy činí 2 minuty. Délka simulace byla stanovena na 250 minut. Výsledkem simulace je posloupnost  $\{x_t, t \in \mathcal{T}\}$ , kde  $x_t = ((x_{A_1})_t, (x_{A_2})_t, (x_{A_3})_t, (x_B)_t)$ . Jednotlivé složky této posloupnosti značí počty zákazníků ve frontách  $A_1, A_2, A_3$  a  $B$  v časech  $t \in \mathcal{T}$ , kde  $\mathcal{T}$  je množina časových okamžiků, v nichž došlo k některé z událostí. Výsledky ve formě grafů vývoje délek jednotlivých front ukazuje obrázek 2.3.



Obrázek 2.3: Vývoj počtu osob ve frontách

V tomto případě byla simulace ukončena při překročení předem stanovené doby simulace, nebyl tak předem znám počet náhodných čísel vygenerovaných během jednoho běhu simulace.

Jinou možností je předem stanovit počet příchozích zákazníků, simulace by tak byla ukončena v okamžiku obslužení posledního zákazníka. Tento typ simulace je z hlediska implementace výhodnější díky předem známému počtu náhodných čísel potřebných k provedení simulace, pro každého zákazníka je potřeba vygenerovat čas příchodu, dobu čerpání pohonných hmot a dobu obsluhy u pokladny.  $\triangle$

## 2.4 Výkonnostní funkce

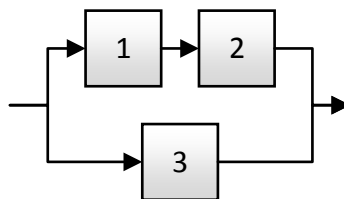
Nechť  $\mathbf{X} \in \Omega$  je stav statického systému reprezentovaný náhodným vektorem. Funkce  $H : \Omega \rightarrow \mathbb{R}$  se nazývá výkonnostní nebo také systémová. Výkonnostní funkce tedy každému stavovému vektoru  $\mathbf{x} \in \Omega$  přiřadí reálné číslo  $y = H(\mathbf{x})$ . Může udávat libovolnou kvalitu systému závislou na náhodném vektoru  $\mathbf{X}$ , viz příklady 2.3 a 2.4.

Podobně v případě simulace dynamického systému v konečném časovém horizontu se jedná o funkci, která každé posloupnosti  $\{x_t, t \in \mathcal{T}\}$  přiřadí reálné číslo  $y = H(\mathbf{x}_t)$ .

**Příklad 2.3.** Uvažujme DESS složený ze tří komponent, jeho strukturu znázorňuje obrázek 2.4. Stav tohoto systému je reprezentován pomocí náhodného vektoru

$$\mathbf{X} = (X_1, X_2, X_3)$$

pocházejícího z vícerozměrného Bernoulliho rozdělení s parametry  $\mathbf{p} = (p_1, p_2, p_3)$ . Jednotlivé složky  $\mathbf{X}$  jsou na sobě nezávislé a  $X_1 \sim \text{Ber}(p_1)$ ,  $X_2 \sim \text{Ber}(p_2)$ ,  $X_3 \sim \text{Ber}(p_3)$ . Náhodná veličina reprezentující stav komponenty může tedy nabýt pouze dvou hodnot, hodnoty 1 (říkáme, že je daná komponenta v provozu) a hodnoty 0 (říkáme, že je daná komponenta mimo provoz).



Obrázek 2.4: Schéma systému z příkladu 2.3

Výkonnostní funkci  $H(\mathbf{X})$  můžeme definovat například jako funkci, která vrací hodnotu 1, je-li systém v provozu (tj. existuje-li cesta ze vstupu na výstup pouze přes funkční komponenty) a hodnotu 0 v opačném případě. Lze snadno ověřit, že tyto podmínky splňuje například funkce

$$H(\mathbf{X}) = 1 - (1 - X_1 X_2)(1 - X_3).$$

Střední hodnota náhodné veličiny  $H(\mathbf{X})$  udává pravděpodobnost, že je systém v provozu, tedy tzv. pohotovost systému.  $\triangle$

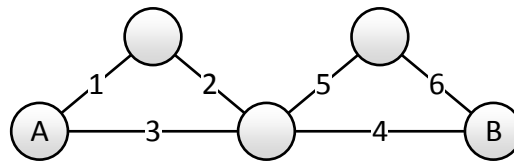
Následuje příklad jiného systému, v němž náhodné veličiny reprezentující stavy jednotlivých komponent pochází ze spojitého rozdělení pravděpodobnosti.

**Příklad 2.4** (Nejkratší cesta, část 1). Uvažujme DESS složený ze šesti komponent reprezentujících cesty, strukturu systému znázorňuje obrázek 2.5. Jednotlivé složky náhodného vektoru

$$\mathbf{X} = (X_1, X_2, X_3, X_4, X_5, X_6),$$

který reprezentuje stav tohoto systému, značí délky jednotlivých cest.





Obrázek 2.5: Schéma systému z příkladu 2.4

Výkonnostní funkci  $S(\mathbf{X})$  lze definovat například jako délku nejkratší cesty z bodu A do bodu B, tj.

$$S(\mathbf{X}) = \min \{X_1 + X_2 + X_4, X_1 + X_2 + X_5 + X_6, X_3 + X_4, X_3 + X_5 + X_6\}.$$

Střední hodnota náhodné veličiny  $S(\mathbf{X})$  tedy udává pravděpodobnou délku nejkratší cesty z bodu A do bodu B.

Jinou možností je výkonnostní funkce  $H(\mathbf{X})$ , která vrací hodnotu 0, je-li délka minimální cesty kratší než daná hodnota  $\gamma \in \mathbb{R}$ , a hodnotu 1 v opačném případě, tj.

$$H(\mathbf{X}) = I_{\{S(\mathbf{X}) \geq \gamma\}}.$$

Střední hodnota náhodné veličiny  $H(\mathbf{X})$  v tomto případě udává pravděpodobnost, že délka nejkratší cesty překročí hodnotu  $\gamma$ .  $\triangle$

### 3 Statistická analýza DE systémů

Tato kapitola se zabývá statistickou analýzou dat získaných simulací statických a dynamických DE systémů. V obou případech je systém reprezentován pomocí stavu systému, kterým je obvykle myšlen náhodný vektor konečné dimenze. V případě dynamických modelů tento vektor navíc závisí na čase a chování systému je tak reprezentováno jako stochastický proces.

#### 3.1 Výsledek simulace

Výsledkem  $N$  opakování simulace DESS je náhodný výběr  $\mathbf{X}_1, \dots, \mathbf{X}_N$  z náhodné veličiny  $\mathbf{X}$ . Aplikací výkonnostní funkce  $H$  na jednotlivé náhodné vektory je získán výběrový soubor  $Y_1 = H(\mathbf{X}_1), \dots, Y_N = H(\mathbf{X}_N)$  z náhodné veličiny  $Y = H(\mathbf{X})$ . Podobně v případě simulace DEDS v konečném časovém horizontu je výsledkem simulace výběrový soubor  $Y_1 = H((\mathbf{X}_t)_1), \dots, Y_N = H((\mathbf{X}_t)_N)$  z náhodné veličiny  $Y = H(\mathbf{X}_t)$ . Prostřednictvím náhodné veličiny  $Y$  reprezentující výsledek simulace lze oba případy ztotožnit.

Prvotní informaci o simulačních datech lze získat jejich vizualizací například pomocí histogramu nebo empirické distribuční funkce. V prostředí Matlab lze snadno vykreslit histogram pomocí příkazu `histogram`, případně ve starších verzích programu Matlab pomocí příkazu `hist`. Pro vykreslení empirické distribuční funkce slouží příkazy `ecdf` a `cdfplot`. Pro odhad hustoty spojitého rozdělení pravděpodobnosti, z něhož pochází výběrový soubor  $Y_1, \dots, Y_N$ , lze dále použít funkci `ksdensity`, viz [4].

Základními charakteristikami výběrového souboru jsou obecné a centrální výběrové momenty, které lze použít k odhadu příslušných charakteristik náhodné veličiny  $Y = H(\mathbf{X})$ . Nejčastějším úkolem statistické analýzy simulačních dat je odhadnout střední hodnotu náhodné veličiny  $Y$  pomocí výběrového průměru náhodného výběru  $Y_1, \dots, Y_N$ , tento proces podrobně popisuje následující sekce.

#### 3.2 Metoda Monte Carlo

Tato sekce se věnuje základní metodě Monte Carlo, využívá teoretických poznatků uvedených například v [5] a [1]. Název MC metody či MC techniky v literatuře obvykle zahrnuje celou skupinu simulačních metod vycházejících ze základní metody popsané v této sekci. Další metody patřící do této skupiny budou představeny v kapitole 4.

Řešme nyní následující obecnou úlohu:

**Úloha 3.1.** Nechť  $\mathbf{X} = (X_1, \dots, X_n) \in \mathbb{R}^n$  je náhodný vektor pocházející z pravděpodobnostního rozdělení o hustotě  $f$  a  $H : \mathbb{R}^n \rightarrow \mathbb{R}$  je výkonnostní funkce. Určete střední hodnotu  $\ell$  náhodné veličiny  $H(\mathbf{X})$ .

Není-li možné vyčíslit hodnotu  $\ell = \mathbb{E}(H(\mathbf{X}))$  analyticky, lze přejít k metodě Monte Carlo a řešit tak úlohu simulačně.

**Věta 3.2** (MC estimátor). Necht'  $Y_1, Y_2, \dots, Y_N$  jsou navzájem nezávislé realizace náhodné veličiny  $Y$ . Pak MC estimátor

$$\hat{\ell} = \frac{1}{N} \sum_{k=1}^N Y_k \quad (3.1)$$

je nestranným odhadem  $\ell = \mathbb{E}Y$  a pro jeho rozptyl platí

$$\text{Var}(\hat{\ell}) = \frac{1}{N} \text{Var}(Y).$$

*Důkaz.* Estimátor  $\hat{\ell}$  je nestranným odhadem  $\ell$ , neboť

$$\mathbb{E}(\hat{\ell}) = \mathbb{E}\left(\frac{1}{N} \sum_{i=1}^N Y_i\right) = \frac{1}{N} \sum_{i=1}^N \mathbb{E}(Y_i) = \mathbb{E}Y = \ell.$$

Pro jeho rozptyl platí

$$\text{Var}(\hat{\ell}) = \text{Var}\left(\frac{1}{N} \sum_{i=1}^N Y_i\right) = \frac{1}{N^2} \sum_{i=1}^N \text{Var} Y = \frac{1}{N} \text{Var} Y.$$

□

Postup řešení úlohy 3.1 metodou MC shrnuje následující algoritmus.

**Algoritmus 3.3** (Metoda MC).

1. Vygenerování náhodného výběru  $\mathbf{X}_1, \dots, \mathbf{X}_N$  z rozdělení pravděpodobnosti o hustotě  $f$ .
2. Výpočet hodnot  $Y_1 = H(\mathbf{X}_1), \dots, Y_N = H(\mathbf{X}_N)$ .
3. Odhad hodnoty  $\ell$  výběrovým průměrem

$$\hat{\ell} = \frac{1}{N} \sum_{k=1}^N Y_k. \quad (3.2)$$

4. Odhad  $(1 - \alpha)$  100% intervalu spolehlivosti pro parametr  $\ell$ .

Součástí postupu řešení je odhad přesnosti získaného odhadu, určíme proto interval spolehlivosti, v němž se s pravděpodobností  $1 - \alpha$  nachází skutečná hodnota  $\ell$ . Hledáme takovou hodnotu  $\varepsilon > 0$ , pro níž platí

$$\mathbb{P}(\ell \in (\hat{\ell} - \varepsilon, \hat{\ell} + \varepsilon)) = 1 - \alpha. \quad (3.3)$$

Podle centrální limitní věty, viz například [6], platí

$$\hat{\ell} \sim \mathcal{N}\left(\ell, \frac{\sigma^2}{N}\right),$$

kde  $\sigma$  značí směrodatnou odchylku náhodné veličiny  $Y$ .

$$\begin{aligned} \mathbb{P}\left(\ell \in \left(\hat{\ell} - \varepsilon, \hat{\ell} + \varepsilon\right)\right) &= \mathbb{P}\left(-\varepsilon < \hat{\ell} - \ell < \varepsilon\right) = \mathbb{P}\left(-\frac{\varepsilon}{\sigma}\sqrt{N} < \frac{\hat{\ell} - \ell}{\sigma}\sqrt{N} < \frac{\varepsilon}{\sigma}\sqrt{N}\right) = \\ &= \phi\left(\frac{\varepsilon}{\sigma}\sqrt{N}\right) - \phi\left(-\frac{\varepsilon}{\sigma}\sqrt{N}\right) = 2\phi\left(\frac{\varepsilon}{\sigma}\sqrt{N}\right) - 1, \end{aligned}$$

což dosadíme do (3.3) a vyjádříme  $\varepsilon$ :

$$\begin{aligned} 2\phi\left(\frac{\varepsilon}{\sigma}\sqrt{N}\right) - 1 &= 1 - \alpha \\ \frac{\varepsilon}{\sigma}\sqrt{N} &= z_{1-\frac{\alpha}{2}} \\ \varepsilon &= \frac{\sigma}{\sqrt{N}} z_{1-\frac{\alpha}{2}} \end{aligned} \tag{3.4}$$

Střední hodnota tedy s pravděpodobností  $1 - \alpha$  leží v intervalu

$$\left(\hat{\ell} - z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{N}}, \hat{\ell} + z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{N}}\right). \tag{3.5}$$

Ze vztahu (3.4) lze rovněž určit počet náhodných pokusů nutných k dosažení předem dané přesnosti  $\varepsilon$  s pravděpodobností  $1 - \alpha$ ,

$$N_\varepsilon = \left(\frac{\sigma}{\varepsilon} z_{1-\frac{\alpha}{2}}\right)^2.$$

Hodnota  $\alpha$  je předem zvoleným koeficientem spolehlivosti, obvykle například  $\alpha = 0.05$ .

Ve výše odvozených vzorcích vystupuje směrodatná odchylka  $\sigma$ , jejíž hodnota obvykle není známa. Směrodatnou odchylku  $\sigma$  je však možné nahradit výběrovou směrodatnou odchylkou

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (Y_i - \hat{\ell})^2},$$

lze snadno ukázat, že  $s$  je nestranným odhadem  $\sigma$ . Přibližným  $(1 - \alpha)$  100% intervalem spolehlivosti pro parametr  $\ell$  je tedy interval

$$\left(\hat{\ell} - z_{1-\frac{\alpha}{2}} \frac{s}{\sqrt{N}}, \hat{\ell} + z_{1-\frac{\alpha}{2}} \frac{s}{\sqrt{N}}\right). \tag{3.6}$$

Mírou přesnosti odhadu  $\hat{\ell}$ , která nezávisí na koeficientu spolehlivosti, je relativní směrodatná odchylka RSD (z anglického *relative standard deviation*) definovaná vztahem

$$\text{RSD} = \frac{s}{\hat{\ell} \cdot \sqrt{N}}.$$

Pro  $N \rightarrow \infty$  konverguje RSD k relativní chybě odhadu RE (z anglického *relative error*), tj.

$$\text{RSD} \xrightarrow{N \rightarrow \infty} \text{RE} = \frac{\sqrt{\text{Var}(\hat{\ell})}}{\mathbb{E}(\hat{\ell})}. \quad (3.7)$$

Veličiny RSD i RE bývají často udávány v procentech.

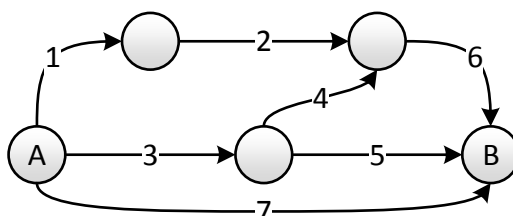
Použití metody MC bude nyní demonstrováno na jednoduchém příkladu. Tento příklad bude dále řešen v kapitole 4 s využitím metod pro redukci rozptylu.

**Příklad 3.4** (PERT, část 1). Technika PERT (z anglického *Program evaluation and review technique*) je nástrojem využívaným při organizaci projektů. Složitější projekt lze obvykle rozdělit na řadu činností, z nichž některé mohou být vykonány zároveň a s některými je možné začít až po dokončení jiných.

Uvažujme konkrétní projekt složený ze sedmi činností. Tabulka 3.1 pro každou činnost uvádí, které z ostatních činností musí být vykonány před jejím zahájením. Přehledněji lze projekt znázornit pomocí tzv. PERT sítě, viz obrázek 3.1. Jedná se o orientovaný graf, v němž každá hrana značí jednu činnost. Orientovaná hrana vedoucí z vrcholu V symbolizuje činnost, která smí být zahájena až po dokončení všech činností symbolizovaných orientovanými hranami vedoucími do vrcholu V. Vrchol A značí začátek projektu, vrchol B konec projektu.

Činnost	1	2	3	4	5	6	7
Předchozí činnosti	-	1	-	3	3	2,4	-

Tabulka 3.1: Tabulka projektu



Obrázek 3.1: PERT síť odpovídající tabulce 3.1

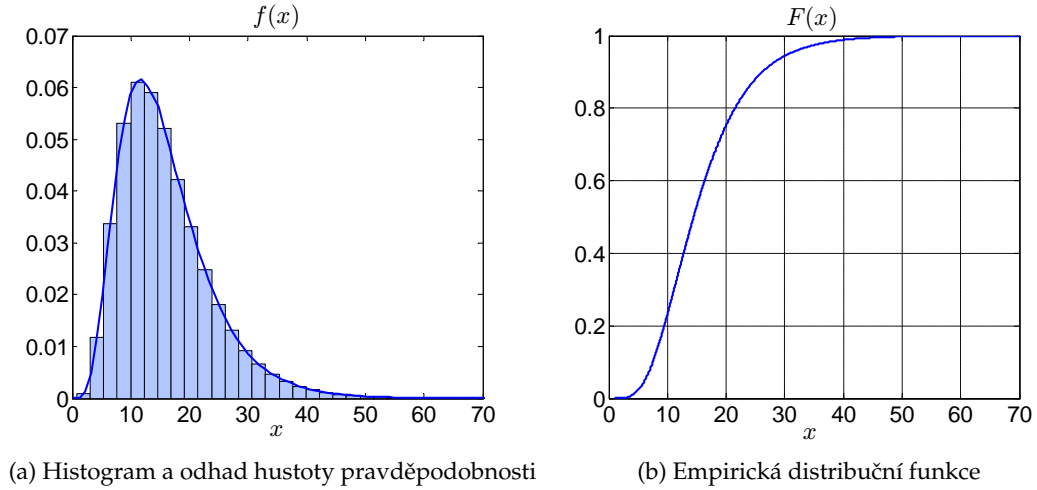
Doby trvání jednotlivých činností jsou náhodnými veličinami z exponenciálního rozdělení. Pro jednoduchost předpokládejme, že činnosti  $i$  přísluší náhodná veličina  $X_i \sim \text{Exp}(i^{-1})$ , tedy střední doba do dokončení činnosti  $i$  je rovna  $i$ .

Projekt je charakterizován zejména odhadovanou dobou jeho trvání. Doba trvání projektu je rovna délce nejdelší cesty z bodu A do bodu B, která je obvykle nazývána kritickou cestou. Výkonnostní funkce  $H(\mathbf{X})$  tedy pro vygenerovaný vektor  $\mathbf{X} = (X_1, \dots, X_7)$  vrací délku kritické cesty, neboli

$$H(\mathbf{X}) = \max \{X_1 + X_2 + X_6, X_3 + X_4 + X_6, X_3 + X_5, X_7\}.$$

Úkolem je odhadnout pravděpodobnou dobu trvání projektu  $\ell = \mathbb{E}(H(\mathbf{X}))$  pomocí metody MC se spolehlivostí  $1 - \alpha = 0.95$ .

Simulace byla provedena pro  $N = 10^6$  náhodných pokusů. Na základě získaného výběrového souboru  $H(\mathbf{X}_1), \dots, H(\mathbf{X}_N)$  byla vykreslena empirická distribuční funkce a odhadnuta hustota pravděpodobnosti náhodné veličiny  $H(\mathbf{X})$ , viz následující obrázek.



Obrázek 3.2: Vizualizace výsledků simulace

Odhadem hledané hodnoty  $\ell$  je výběrový průměr

$$\hat{\ell} = \frac{1}{N} \sum_{k=1}^N H(\mathbf{X}_k) \doteq 15.926$$

a odhadem směrodatné odchylky  $\sigma$  výběrová směrodatná odchylka  $s \doteq 7.827$ , která byla použita pro odhad intervalu spolehlivosti pro parametr  $\ell$ . S pravděpodobností  $1 - \alpha$

$$\ell \in \left( \hat{\ell} - z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{N}}, \hat{\ell} + z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{N}} \right) \approx (15.911, 15.942).$$

Pro dosažení přesnosti  $\varepsilon = 0.01$  s pravděpodobností  $1 - \alpha$  by bylo třeba provést

$$N_\varepsilon = \left( \frac{\sigma}{\varepsilon} z_{1-\frac{\alpha}{2}} \right)^2 \approx 2353340$$

náhodných pokusů.

Tabulka 3.2 zobrazuje zmíněné výsledky a slouží k porovnání s výsledky téže úlohy, které budou získány v kapitole 4 pomocí metod redukce rozptylu. Tato úloha bude vždy řešena pomocí simulace o rozsahu  $N = 10^6$  a pro vyhodnocení budou použity parametry  $\alpha = 0.05$  a  $\varepsilon = 0.01$ .

$\hat{\ell}$	$s$	$s^2$	RSD	interval spolehlivosti	$N_\varepsilon$
15.926	7.827	61.262	$4.91 \cdot 10^{-2}\%$	(15.911, 15.942)	2353340

Tabulka 3.2: Výsledky příkladu 3.4 získané metodou MC

△

Následující příklad ilustruje použití metody MC jako nástroje k vyhodnocení dat získaných při simulaci DEDS v konečném časovém horizontu.

**Příklad 3.5** (Systém hromadné obsluhy, část 2). Jedná se o systém hromadné obsluhy představený v příkladu 2.2 v předchozí kapitole. Simulace byla provedena pro  $k = 6$  front reprezentujících stojany čerpací stanice. Počet příchozích zákazníků byl předem stanoven na 200, doba do příjezdu zákazníka pochází z Exp (1). Doba čerpání pohonných hmot pochází z Weib  $(3 \cdot (\Gamma(1.5))^{-1}, 2)$ , čerpání benzínu tedy trvá v průměru 3 minuty, podobně doba obsluhy u pokladny pochází z Weib  $(0.5 \cdot (\Gamma(1.5))^{-1}, 2)$ .

Na základě simulace o rozsahu  $N = 10^4$  byla stanovena průměrná doba, kterou zákazník stráví na čerpací stanici, a určen průměrný počet osob ve frontách  $A_1, \dots, A_6$  a ve frontě  $B$ .

Odhadem průměrné doby  $\ell$  mezi příjezdem zákazníka a jeho odjezdem z čerpací stanice je  $\hat{\ell} = 3.911$  a pro výběrový rozptyl odhadu platí

$$s^2 = 3.38 \cdot 10^{-2},$$

odhadem 95% intervalu spolehlivosti pro parametr  $\ell$  je tedy interval (3.907, 3.914).

Průměrný počet osob ve frontách znázorňují grafy na obrázku 3.3, graf 3.3a ukazuje pravděpodobný vývoj součtu osob ve frontách  $A_1, \dots, A_6$  v čase a graf 3.3b pravděpodobný vývoj počtu osob ve frontě  $B$ .

△

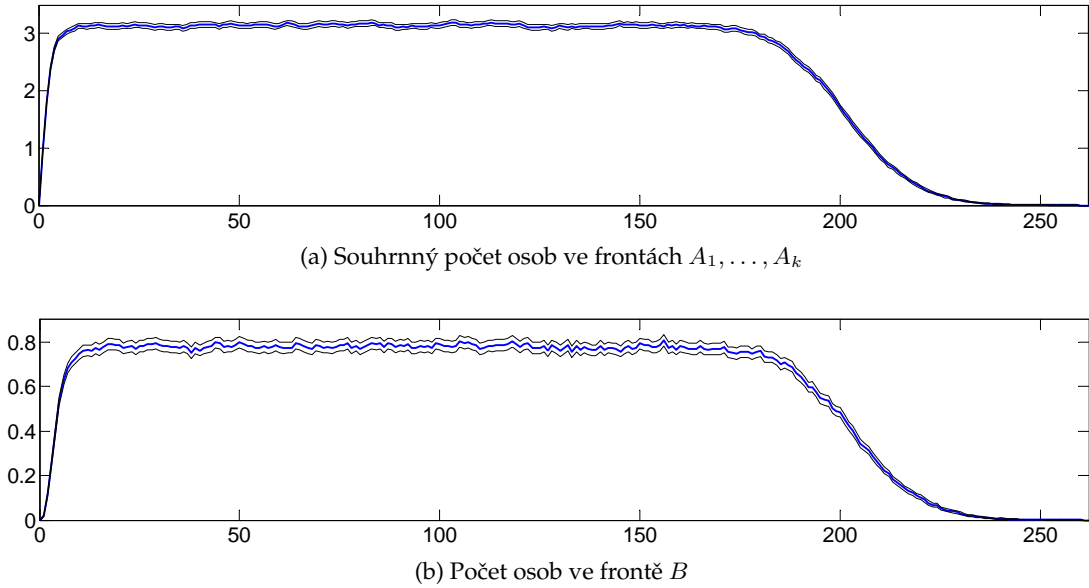
Jestliže náhodná veličina  $H(\mathbf{X})$  nabývá pouze hodnot 0 a 1, lze algoritmus 3.3 zjednodušit. Stačí provést  $N$  pokusů a zaznamenat počet pokusů  $N_1$ , ve kterých nabyla NV  $H(\mathbf{X})$  hodnotu 1. Nestranným odhadem hodnoty  $\ell$  je pak hodnota

$$\hat{\ell} = \frac{N_1}{N}$$

a pro výběrovou směrodatnou odchylku  $s$  lze s využitím poznatku  $(H(\mathbf{X}))^2 = H(\mathbf{X})$  snadno odvodit vzorec

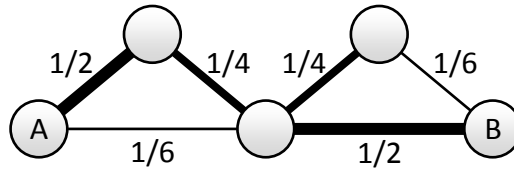
$$s = \sqrt{\frac{N_1}{N-1} (1 - \hat{\ell})}.$$

Pomocí metody MC bude dále řešen příklad 2.4, formulovaný v předchozí kapitole.



Obrázek 3.3: Vývoj počtu osob ve frontách

**Příklad 3.6** (Nejkratší cesta, část 2). Uvažujme systém znázorněný na obrázku 2.5. Předpokládejme, že náhodný vektor  $\mathbf{X}$ , reprezentující délky jednotlivých cest, pochází z Weibullova rozdělení s parametrem tvaru  $\alpha = 2$  a parametrem měřítka  $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_6) = (2^{-1}, 4^{-1}, 6^{-1}, 2^{-1}, 4^{-1}, 6^{-1})$ .

Obrázek 3.4: Schéma systému z příkladu 3.6 se znázorněním parametrů  $\lambda_i$ 

Úkolem je pomocí metody MC určit hodnotu  $\ell = \mathbb{E}(H(\mathbf{X}))$ , tedy pravděpodobnost, že délka nejkratší cesty překročí hodnotu  $\gamma = 16$ .

Simulace byla provedena pro  $N = 10^9$ , výsledkem je hodnota  $N_1 = 5307$ . Odhadem hodnoty  $\ell$  je tedy  $\hat{\ell} = 5.307 \cdot 10^{-6}$ . Odhadem 95% intervalu pro parametr  $\ell$  je interval

$$(5.164 \cdot 10^{-6}, 5.45 \cdot 10^{-6}).$$

Pro dosažení přesnosti  $\varepsilon = 10^{-7}$  s pravděpodobností 95% by muselo být provedeno přibližně  $N_\varepsilon = 2.039 \cdot 10^9$  náhodných pokusů.  $\triangle$



### 3.3 Řídké jevy

Řídký jev (anglicky *rare event*) je událost, ke které dochází s velmi nízkou pravděpodobností, podle [2, str. 382] se jedná o pravděpodobnost nižší než  $10^{-4}$ . Pro řídký jev  $A$  tedy platí

$$\ell = \mathbb{P}(A) = \mathbb{E}(I_A) < 10^{-4}.$$

Hodnota  $\ell$  v tomto případě udává pravděpodobnost, že nastane řídký jev  $A$ .

Nechť dále  $Z$  je náhodná veličina, pro níž platí  $\mathbb{E}(Z) = \ell$ , a  $Z_1, \dots, Z_N$  jsou navzájem nezávislé realizace  $Z$ , pak

$$\hat{\ell} = \frac{1}{N} \sum_{i=1}^N Z_i \quad (3.8)$$

je nestranným odhadem pravděpodobnosti  $\ell$ . Je-li  $Z = I_A$ , jedná se přímo o MC estimátor (3.1).

Jev  $A$  lze obvykle zapsat ve tvaru  $A = \{S(\mathbf{X}) \geq \gamma\}$ , kde  $S : \mathbb{R}^n \rightarrow \mathbb{R}$  je výkonnostní funkce daného systému a úroveň  $\gamma$  je hodnota, kterou tato výkonnostní funkce překročí s velmi nízkou pravděpodobností. Obecná úloha 3.1 pak přechází do následujícího podoby:

**Úloha 3.7** (Kvantifikace řídkých jevů). Nechť  $\mathbf{X} = (X_1, \dots, X_n) \in \mathbb{R}^n$  je náhodný vektor pocházející z rozdělení pravděpodobnosti o hustotě  $f$ ,  $S : \mathbb{R}^n \rightarrow \mathbb{R}$  je výkonnostní funkce a  $\gamma \in \mathbb{R}$ . Určete střední hodnotu  $\ell$  náhodné veličiny  $H(\mathbf{X}) = I_{\{S(\mathbf{X}) \geq \gamma\}}$ .

Pravděpodobnost  $\ell$  lze v tomto případě chápat jako funkci parametru  $\gamma$ , tedy

$$\ell = \ell(\gamma) = \mathbb{P}(S(\mathbf{X}) \geq \gamma) = \mathbb{E}(I_{\{S(\mathbf{X}) \geq \gamma\}}),$$

přičemž  $\ell(\gamma) \rightarrow 0$  pro  $\gamma \rightarrow \infty$ . Obdobně lze chápat nestranný odhad  $\hat{\ell}$  a náhodnou veličinu  $Z$  jako funkce parametru  $\gamma$ , tj.  $\hat{\ell} = \hat{\ell}(\gamma)$  a  $Z = Z(\gamma)$ .

**Příklad 3.8.** Předpokládejme, že pravděpodobnost nastání jisté události je rovna hodnotě  $\ell$ . Tato hodnota je odhadována pomocí MC estimátoru. Určeme nyní počet pokusů  $N_\varepsilon$  nutných k dosažení přesnosti  $\varepsilon = 0.1 \cdot \ell$  s pravděpodobností 95%.

Jelikož  $\sigma^2 = \ell - \ell^2$ , platí

$$N_\varepsilon = (\ell - \ell^2) \left( \frac{1}{\varepsilon} z_{1-\frac{\alpha}{2}} \right)^2 \doteq 384 \cdot \left( \frac{1}{\ell} - 1 \right).$$

Již pro  $\ell = 10^{-4}$  je tedy potřeba provést více než  $3.8 \cdot 10^6$  náhodných pokusů.  $\triangle$

K jevu „délka nejkratší cesty překročí hodnotu  $\gamma$ “, zaznamenanému v příkladu 3.6, dochází s pravděpodobností přibližně  $5.3 \cdot 10^{-6}$ , jde tedy o řídký jev. Jelikož se jednalo o jednoduchý statický systém, bylo možné provést vysoký počet náhodných pokusů a dosáhnout tak poměrně přesného výsledku. Nicméně v případě komplikovaných systémů obvykle není z časových důvodů možné provést simulaci o dostatečném rozsahu. Nutností se tak stává použití tzv. metod redukce rozptylu, které dosáhnou téže přesnosti při výrazně nižším počtu náhodných pokusů.

## 4 Přehled metod pro redukci rozptylu

V předchozí kapitole byla popsána metoda MC jako nástroj k analýze DE systémů. Rovněž bylo zmíněno několik způsobů vyhodnocení přesnosti odhadu získaného metodou MC. Cílem této kapitoly je představit několik metod redukce rozptylu, které vycházejí z MC, ale jsou efektivnější z pohledu přesnosti získaného odhadu. Tyto metody budou opět aplikovány na obecný problém 3.1, hodnota  $\ell$  bude vždy odhadována výběrovým průměrem z náhodné veličiny, jejíž střední hodnota je rovna hledané hodnotě  $\ell$ . Ve všech případech se tedy bude jednat o nestranný odhad. Tato kapitola čerpá především z [1, 2, 8].

Nejpoužívanější z dále uvedených metod je zejména díky značné efektivitě dosahované při odhadu pravděpodobností řídkých jevů technika *importance sampling*. Této metodě a jejím variantám je tedy věnována největší pozornost.

Definice 4.1 umožňuje sjednocení pojmů pravděpodobnostní funkce diskrétní NV a hustota pravděpodobnosti spojitě NV, pochází z [8, str. 3] a z ní vycházející značení je používáno i v dalších publikacích, například [1, 2]. Mnohdy je místo  $\mu(dx)$  psáno pro zjednodušení pouze  $dx$ .

**Definice 4.1.** Rozdělení pravděpodobnosti náhodného vektoru  $\mathbf{X}$  je chápáno jako míra  $\nu(B) = \mathbb{P}(\mathbf{X} \in B)$ , kde  $B \in \mathcal{B}$ . (Symbol  $\mathcal{B}$  značí  $\sigma$ -algebru borelovských množin nad  $\mathbb{R}^n$ , viz [9].)

Je-li  $f$  pravděpodobnostní funkcí diskrétního náhodného vektoru  $\mathbf{X}$ , nebo hustotou pravděpodobnosti spojitě náhodného vektoru  $\mathbf{X}$ , platí

$$\nu(B) = \int_B f(\mathbf{x}) \mu(d\mathbf{x}).$$

V diskrétním případě je  $\mu$  aritmetickou mírou a ve spojitěm případě se jedná o Lebesgueovu míru. Funkce  $f$  bude dále ve spojitěm i diskrétním případě nazývána hustotou pravděpodobnosti a střední hodnota náhodné veličiny  $g(\mathbf{X})$ , kde  $g$  je měřitelná funkce, bude zapisována jako

$$\mathbb{E}(g(\mathbf{X})) = \int g(\mathbf{x}) f(\mathbf{x}) \mu(d\mathbf{x}).$$

### 4.1 Metoda antitetických náhodných veličin

**Definice 4.2.** Náhodné veličiny  $Y$  a  $Y^*$  tvoří **antitetickou dvojici**, jestliže pochází ze stejného rozdělení a jsou negativně korelované.

Metoda antitetických náhodných veličin je založena na následující větě.

**Věta 4.3** (Antitetický estimátor). *Necht'*

- $Y$  je náhodná veličina,
- $(Y_1, Y_1^*), \dots, (Y_{N/2}, Y_{N/2}^*)$ , kde  $N$  je sudé, jsou nezávislé antitetické dvojice,
- $\forall k \in \{1, \dots, \frac{N}{2}\}$  mají náhodné veličiny  $Y_i$  a  $Y_i^*$  stejné rozdělení jako  $Y$ .

*Pak antitetický estimátor*

$$\hat{\ell}^{(a)} = \frac{1}{N} \sum_{k=1}^{N/2} (Y_k + Y_k^*) \quad (4.1)$$

*je nestranným odhadem  $\ell = \mathbb{E}Y$  a pro jeho rozptyl platí*

$$\text{Var}(\hat{\ell}^{(a)}) = \frac{\text{Var}(Y)}{N} (1 + \rho_{Y,Y^*}).$$

*Důkaz.* Pro střední hodnotu estimátoru platí

$$\mathbb{E}(\hat{\ell}^{(a)}) = \mathbb{E}\left(\frac{1}{N} \sum_{k=1}^{N/2} (Y_k + Y_k^*)\right) = \frac{1}{N} \sum_{k=1}^{N/2} (\mathbb{E}Y_k + \mathbb{E}Y_k^*) = \frac{1}{N} \sum_{k=1}^{N/2} (2 \cdot \mathbb{E}Y) = \mathbb{E}Y = \ell,$$

$\hat{\ell}^{(a)}$  je tedy nestranným odhadem  $\ell$ . Pro dokončení důkazu zbývá určit rozptyl estimátoru,

$$\begin{aligned} \text{Var}(\hat{\ell}^{(a)}) &= \text{Var}\left(\frac{1}{N} \sum_{k=1}^{N/2} (Y_k + Y_k^*)\right) = \frac{1}{N^2} \sum_{k=1}^{N/2} (\text{Var} Y_k + \text{Var} Y_k^* + 2 \text{Cov}(Y_k, Y_k^*)) = \\ &= \frac{1}{N} (\text{Var} Y + \text{Cov}(Y, Y^*)) = \frac{\text{Var}(Y)}{N} (1 + \rho_{Y,Y^*}). \end{aligned}$$

□

Podle věty 4.3 je rozptyl odhadu  $\hat{\ell}^{(a)}$  přímo úměrný hodnotě  $1 + \rho_{Y,Y^*}$ , přičemž  $\rho_{Y,Y^*} \in (-1, 0)$ . Největší efektivity tedy metoda antitetických náhodných veličin dosáhne, pokud budou náhodné veličiny  $Y$  a  $Y^*$  silně záporně korelované. Zbývá poznamenat, že jistého snížení výpočetní náročnosti oproti metodě MC je dosaženo rovněž díky polovičnímu počtu generovaných náhodných čísel.

Řešme nyní obecnou úlohu 3.1 a předpokládejme navíc, že funkce  $H : \mathbb{R}^n \rightarrow \mathbb{R}$  je monotónní v každé složce a že jednotlivé složky náhodného vektoru  $\mathbf{X}$  pochází z rovnoměrného rozdělení  $\mathcal{U}(0, 1)$ . Následuje algoritmus řešení úlohy 3.1 s těmito dodatečnými předpoklady pomocí metody antitetických náhodných veličin.

---

**Algoritmus 4.4** (Metoda antitetických náhodných veličin).

1. Vygenerování náhodného výběru  $\mathbf{X}_1, \dots, \mathbf{X}_{\frac{N}{2}}$  z  $\mathcal{U}(0, 1)$ .
2. Výpočet hodnot  $Y_1 = H(\mathbf{X}_1), \dots, Y_{\frac{N}{2}} = H(\mathbf{X}_{\frac{N}{2}})$  a  $Y_1^* = H(\mathbf{1} - \mathbf{X}_1), \dots, Y_{\frac{N}{2}}^* = H(\mathbf{1} - \mathbf{X}_{\frac{N}{2}})$ .
3. Odhad hodnoty  $\ell$  výběrovým průměrem z náhodné veličiny

$$\frac{Y + Y^*}{2},$$

tedy nestranným odhadem (4.1).

4. Odhad  $(1 - \alpha)$  100% intervalu spolehlivosti pro parametr  $\ell$ .
- 

Podle (3.5) je  $(1 - \alpha)$  100% intervalem spolehlivosti pro parametr  $\ell$  interval

$$\left( \widehat{\ell}^{(a)} - z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{\frac{N}{2}}}, \widehat{\ell}^{(a)} + z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{\frac{N}{2}}} \right).$$

Zbývá tedy určit směrodatnou odchylku  $\sigma$  jako odmocninu z rozptylu,

$$\begin{aligned} \text{Var} \left( \frac{1}{2} (Y + Y^*) \right) &= \frac{1}{4} \text{Var} (Y + Y^*) = \frac{1}{4} (\text{Var} Y + \text{Var} Y^* + 2 \text{Cov} (Y, Y^*)) = \\ &= \frac{1}{4} (\text{Var} Y + \text{Var} Y^* + 2 \text{Cov} (Y, Y^*)), \end{aligned}$$

$$\sigma = \sqrt{\text{Var} \left( \frac{1}{2} (Y + Y^*) \right)} = \frac{1}{2} \sqrt{\text{Var} Y + \text{Var} Y^* + 2 \text{Cov} (Y, Y^*)}.$$

Podobně jako v případě (3.6) lze neznámé hodnoty rozptylu a kovariance nahradit hodnotami příslušných výběrových charakteristik a určit tak odhad intervalu spolehlivosti.

Metod, jak zajistit, aby náhodné veličiny  $Y$  a  $Y^*$  tvořily antitetickou dvojici, existuje více, například lze využít symetrii normálního rozdělení. Jestliže  $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$  a  $H$  je monotónní v každé složce, lze dokázat, že  $Y = H(\mathbf{X})$  a  $Y^* = H(2\boldsymbol{\mu} - \mathbf{X})$  tvoří antitetickou dvojici, viz [7].

Následuje příklad, jehož zadání je shodné s příkladem 3.4, tentokrát je však řešení pomocí metody antitetických náhodných veličin.

**Příklad 4.5** (PERT, část 2). Úkolem je nalézt s použitím algoritmu 4.4 odhad hodnoty

$$\ell = \mathbb{E}(H(\mathbf{X})) = \mathbb{E}(\max\{X_1 + X_2 + X_6, X_3 + X_4 + X_6, X_3 + X_5, X_7\})$$

se spolehlivostí  $1 - \alpha = 0.95$ . Je snadné ověřit, že funkce  $H$  je v každé složce neklesající. Algoritmus 4.4 dále pracuje s náhodnou veličinou z vícerozměrného rovnoměrného rozdělení, jednotlivé složky náhodného vektoru  $\mathbf{X}$  ale pochází z exponenciálního rozdělení. Tento problém lze snadno vyřešit pomocí inverzní transformace.

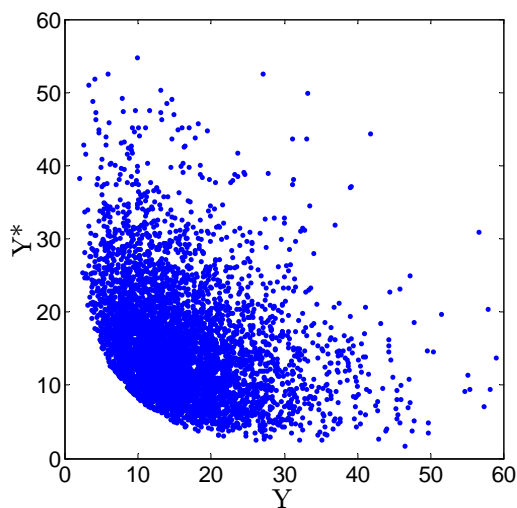
Opět bylo zvoleno  $N = 10^6$ , bylo tedy provedeno  $\frac{N}{2}$  náhodných pokusů. Pro všechna  $k \in \{1, \dots, \frac{N}{2}\}$  byl nejprve vygenerován náhodný vektor  $\mathbf{U}_k$ , jehož složky pochází z  $\mathcal{U}(0, 1)$ , a následně vypočteny hodnoty  $Y_k = H(F^{-1}(\mathbf{U}_k))$  a  $Y_k^* = H(F^{-1}(\mathbf{1} - \mathbf{U}_k))$ . Odhadem hodnoty  $\ell$  je výběrový průměr

$$\widehat{\ell}^{(a)} = \frac{1}{N} \sum_{k=1}^{N/2} (Y_k + Y_k^*) \doteq 15.937.$$

Další výsledky jsou zaznamenány v tabulce 4.1. Výběrový rozptyl náhodného výběru z náhodné veličiny  $\frac{1}{2}(Y + Y^*)$  je přibližně 1.5krát nižší než odpovídající výběrový rozptyl v případě metody MC, pro dosažení stejné přesnosti odhadu tedy stačí provést 1.5krát méně náhodných pokusů.

$\widehat{\ell}^{(a)}$	$s$	$s^2$	RSD	interval spolehlivosti	$N_\varepsilon$
15.937	6.491	42.1365	$4.08 \cdot 10^{-2}\%$	(15.924, 15.950)	1618654

Tabulka 4.1: Výsledky příkladu 4.5 pro  $N = 10^6$ ,  $\alpha = 0.05$ ,  $\varepsilon = 0.01$



Obrázek 4.1: Závislost náhodných veličin  $Y$  a  $Y^*$ , náhodný výběr o rozsahu  $10^4$

Efektivita metody antitetických náhodných veličin závisí na korelačním koeficientu  $\rho_{Y,Y^*}$ , jeho bodovým odhadem je výběrový korelační koeficient  $r_{Y,Y^*}$ , zde

$$r_{Y,Y^*} = -0.314.$$

Závislost náhodných veličin  $Y, Y^*$  ilustruje rovněž obrázek 4.1. Jednotlivé body grafu mají souřadnice  $[H(F^{-1}(U_k)), H(F^{-1}(1 - U_k))], k = 1, \dots, 10^4$ .  $\triangle$

## 4.2 Metoda řídicích veličin

**Definice 4.6.** Náhodná veličina  $C$  je **řídicí veličinou** náhodné veličiny  $Y$ , jestliže  $C$  a  $Y$  jsou korelované a střední hodnota náhodné veličiny  $C$  je známa.

Metoda řídicích veličin (anglicky *control variables method*) vychází z věty 4.7. Jedná se opět o metodu odhadu  $\ell = \mathbb{E}(Y)$ , její princip spočívá ve využití řídicí veličiny se známou střední hodnotou ke konstrukci náhodné veličiny, která je rovněž nestranným odhadem hodnoty  $\ell$ , ale jejíž rozptyl je nižší než rozptyl  $Y$ .

**Věta 4.7.** *Nechť*

- $Y$  je náhodná veličina a  $C$  je řídicí veličinou  $Y$ ,
- $Y_1, Y_2, \dots, Y_N$  jsou navzájem nezávislé realizace náhodné veličiny  $Y$ ,
- $C_1, C_2, \dots, C_N$  jsou odpovídající realizace řídicí veličiny  $C$ ,
- $\mathbb{E}(C) = \ell^{(c)}$ .

*Pak estimátor*

$$\hat{\ell}^{(c)} = \frac{1}{N} \sum_{k=1}^N \left( Y_k - \gamma (C_k - \ell^{(c)}) \right), \quad (4.2)$$

*kde  $\gamma \in \mathbb{R}$ , je nestranným odhadem  $\ell = \mathbb{E}(Y)$  a pro jeho rozptyl platí*

$$\min_{\gamma \in \mathbb{R}} \text{Var}(\hat{\ell}^{(c)}) = \frac{\text{Var}(Y)}{N} (1 - \rho_{Y,C}^2),$$

*přičemž*

$$\operatorname{argmin}_{\gamma \in \mathbb{R}} \text{Var}(\hat{\ell}^{(c)}) = \frac{\text{Cov}(Y, C)}{\text{Var}(C)}.$$

**Důkaz.** Pro střední hodnotu estimátoru platí

$$\mathbb{E}(\hat{\ell}^{(c)}) = \mathbb{E} \left( \frac{1}{N} \sum_{k=1}^N \left( Y_k - \gamma (C_k - \ell^{(c)}) \right) \right) = \frac{1}{N} \sum_{k=1}^N \mathbb{E}(Y) = \mathbb{E}(Y),$$

$\hat{\ell}^{(c)}$  je tedy nestranným odhadem  $\ell$ .

Pro rozptyl estimátoru platí

$$\begin{aligned}\text{Var}(\tilde{\ell}^{(c)}) &= \text{Var}\left(\frac{1}{N}\sum_{k=1}^N(Y_k - \gamma(C_k - \ell^{(c)}))\right) = \\ &= \frac{1}{N^2}\sum_{k=1}^N(\text{Var} Y_k + \gamma^2 \text{Var}(C_k) - 2\gamma \text{Cov}(Y_k, C_k)) = \\ &= \frac{1}{N}(\text{Var} Y + \gamma^2 \text{Var}(C) - 2\gamma \text{Cov}(Y, C)).\end{aligned}$$

Hodnotu  $\tilde{\gamma}$ , která minimalizuje  $\text{Var}(\tilde{\ell}^{(c)})$ , lze nyní určit snadno,

$$\frac{d}{d\gamma} \left[ \frac{1}{N} (\text{Var} Y + \gamma^2 \text{Var}(C) - 2\gamma \text{Cov}(Y, C)) \right] = 0 \Rightarrow \tilde{\gamma} = \frac{\text{Cov}(Y, C)}{\text{Var}(C)}.$$

Minimální hodnotou rozptylu estimátoru je tedy

$$\begin{aligned}\min_{\gamma \in \mathbb{R}} \text{Var}(\tilde{\ell}^{(c)}) &= \frac{1}{N} (\text{Var} Y + \tilde{\gamma}^2 \text{Var}(C) - 2\tilde{\gamma} \text{Cov}(Y, C)) = \\ &= \frac{1}{N} \left( \text{Var} Y + \left( \frac{\text{Cov}(Y, C)}{\text{Var}(C)} \right)^2 \text{Var}(C) - 2 \frac{\text{Cov}(Y, C)}{\text{Var}(C)} \text{Cov}(Y, C) \right) = \\ &= \frac{1}{N} \left( 1 - \frac{(\text{Cov}(Y, C))^2}{\text{Var}(C) \text{Var} Y} \right) \text{Var} Y = \frac{\text{Var} Y}{N} (1 - \rho_{Y,C}^2)\end{aligned}$$

□

Rozptyl estimátoru  $\tilde{\ell}^{(c)}$  je přímo úměrný hodnotě  $1 - \rho_{Y,C}^2$ , čím je tedy korelace mezi náhodnými veličinami  $Y$  a  $C$  silnější, tím větší efektivity metoda řídicích veličin dosáhne.

Algoritmus řešení obecné úlohy 3.1 pomocí metody řídicích veličin je následující:

---

**Algoritmus 4.8** (Metoda řídicích veličin).

1. Vygenerování náhodného výběru  $\mathbf{X}_1, \dots, \mathbf{X}_N$  z rozdělení pravděpodobnosti o hustotě  $f$ .
2. Výpočet hodnot  $Y_1 = H(\mathbf{X}_1), \dots, Y_N = H(\mathbf{X}_N)$  a  $C_1 = H^{(c)}(\mathbf{X}_1), \dots, C_N = H^{(c)}(\mathbf{X}_N)$ .
3. Stanovení odhadu  $\gamma$  optimálního koeficientu  $\tilde{\gamma}$  pomocí výběrové kovariance a výběrového rozptylu.
4. Odhad hodnoty  $\ell$  výběrovým průměrem z náhodné veličiny

$$Y - \gamma(C - \ell^{(c)}),$$

tedy nestranným odhadem (4.2).

5. Odhad  $(1 - \alpha)$  100% intervalu spolehlivosti pro parametr  $\ell$ .

Podle (3.5) je  $(1 - \alpha)$  100% intervalem spolehlivosti pro parametr  $\ell$  interval

$$\left( \widehat{\ell}^{(c)} - z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{N}}, \widehat{\ell}^{(c)} + z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{N}} \right).$$

Směrodatná odchylka  $\sigma$  je odmocninou rozptylu náhodné veličiny  $Y - \gamma (C - \ell^{(c)})$ ,

$$\begin{aligned} \sigma^2 &= \text{Var}(Y) + \gamma^2 \text{Var}(C) - 2\gamma \text{Cov}(Y, C) = \\ &= \text{Var}(Y) + \left( \frac{\text{Cov}(Y, C)}{\text{Var}(C)} \right)^2 \text{Var}(C) - 2 \frac{(\text{Cov}(Y, C))^2}{\text{Var}(C)} = \\ &= \text{Var}(Y) - \frac{\text{Cov}^2(Y, C)}{\text{Var}(C)}. \end{aligned}$$

Pro získání odhadu směrodatné odchylky stačí opět nahradit rozptyl a kovarianci příslušnými výběrovými charakteristikami.

*Poznámka 4.9.* Nechť náhodná veličina  $Y$  je nestranným odhadem hodnoty  $\ell$ . K sestavení náhodné veličiny  $Y_\gamma$ , která je nestranným odhadem  $\ell$  a má zároveň menší rozptyl než náhodná veličina  $Y$ , lze rovněž použít více řídicích veličin. Za předpokladu použití  $m > 1$  řídicích veličin přejde lineární vztah pro konstrukci náhodné veličiny  $Y_\gamma$  do tvaru

$$Y_\gamma = Y - \gamma^T (C - \ell^{(c)}), \quad (4.3)$$

kde  $\gamma \in \mathbb{R}^m$ ,  $C = (C_1, \dots, C_m)^T$  je vektor řídicích veličin a  $\ell^{(c)}$  je vektor středních hodnot jednotlivých řídicích veličin v odpovídajícím pořadí. Náhodná veličina  $Y_\gamma$  je zřejmě nestranným odhadem  $\ell$  a podobně jako v případě jedné řídicí veličiny lze dokázat, že

$$\min_{\gamma \in \mathbb{R}^m} \text{Var}(Y_\gamma) = (1 - R_{Y,C}^2) \text{Var}(Y)$$

a

$$\arg\min_{\gamma \in \mathbb{R}^m} \text{Var}(Y_\gamma) = \Sigma_C^{-1} \sigma_{Y,C},$$

kde  $\sigma_{Y,C} = (\sigma_{Y,C_1}, \dots, \sigma_{Y,C_m})^T$ ,  $\Sigma_C \in \mathbb{R}^{m \times m}$  značí kovarianční matici náhodného vektoru  $C$  a

$$R_{Y,C}^2 = \frac{1}{\text{Var}(Y)} (\sigma_{Y,C})^T \Sigma_C^{-1} \sigma_{Y,C}.$$

Následuje příklad se stejným zadáním jako 3.4, řešený pomocí metody řídicích veličin.

**Příklad 4.10** (PERT, část 3). Hledáme odhad hodnoty

$$\ell = \mathbb{E}(H(\mathbf{X})) = \mathbb{E}(\max\{X_1 + X_2 + X_6, X_3 + X_4 + X_6, X_3 + X_5, X_7\})$$



se spolehlivostí  $1 - \alpha = 0.95$ . Náhodná veličina  $Y = H(\mathbf{X})$  je délkou kritické cesty. Porovnáme-li očekávané délky jednotlivých cest z bodu A do bodu B, lze předpokládat, že s vysokou pravděpodobností bude kritickou cestou cesta tvořená činnostmi 3, 4 a 6. Dá se tedy očekávat silná korelace mezi délkou kritické cesty a délkou cesty tvořené činnostmi 3, 4 a 6. Jako řídicí veličinu náhodné veličiny  $Y$  tedy zvolíme náhodnou veličinu

$$C = H^{(c)}(\mathbf{X}) = X_3 + X_4 + X_6.$$

Jelikož  $\mathbb{E}(X_3) = 3$ ,  $\mathbb{E}(X_4) = 4$  a  $\mathbb{E}(X_6) = 6$ , pro střední hodnotu náhodné veličiny  $C$  platí

$$\ell^{(c)} = \mathbb{E}(C) = \mathbb{E}(X_3 + X_4 + X_6) = \mathbb{E}(X_3) + \mathbb{E}(X_4) + \mathbb{E}(X_6) = 3 + 4 + 6 = 13.$$

Nyní již lze přejít k simulaci. Nejprve vygenerujeme náhodný výběr  $\mathbf{X}_1, \dots, \mathbf{X}_N$ , kde  $N = 10^6$ . Z tohoto náhodného výběru dále vypočteme hodnoty  $Y_1 = H(\mathbf{X}_1), \dots, Y_N = H(\mathbf{X}_N)$  a  $C_1 = H^{(c)}(\mathbf{X}_1), \dots, C_N = H^{(c)}(\mathbf{X}_N)$  a dále určíme odhad  $\gamma$  ideálního parametru  $\tilde{\gamma}$  pomocí výběrových charakteristik, zde

$$\gamma = \frac{q_{Y,C}}{s_C^2} \doteq 0.779,$$

kde  $q_{Y,C}$  značí výběrovou kovarianci  $Y$  a  $C$  a  $s_C^2$  značí výběrový rozptyl  $C$ . Odhad hodnoty  $\ell$  stanovíme jako

$$\hat{\ell}^{(c)} = \frac{1}{N} \sum_{k=1}^N \left( Y_k - \gamma (C_k - \ell^{(c)}) \right) \doteq 15.933.$$

Ostatní výsledky jsou uvedeny v tabulce 4.2. Výběrový rozptyl náhodné náhodného výběru z  $Y - \gamma (C - \ell^{(c)})$  je přibližně 2.5krát nižší než odpovídající výběrový rozptyl v případě metody MC, pro dosažení stejné přesnosti odhadu tedy stačí provést 2.5krát méně náhodných pokusů.

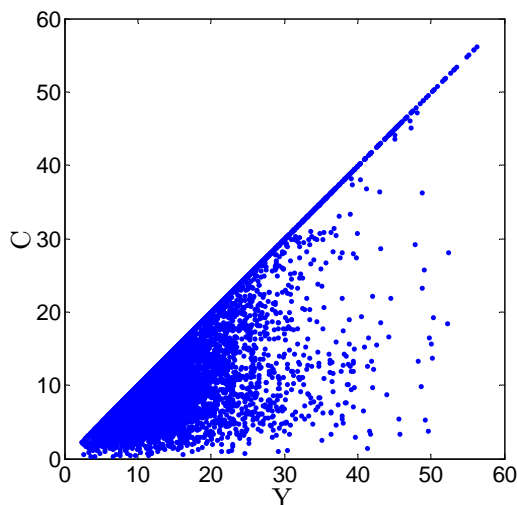
dimenze $C$	$\hat{\ell}^{(c)}$	$s$	$s^2$	RSD	interval spolehlivosti	$N_\varepsilon$
1	15.936	4.918	24.187	0.031	(15.926, 15.945)	929138
4	15.932	3.215	10.336	0.020	(15.926, 15.939)	397037

Tabulka 4.2: Výsledky příkladu 4.5 pro  $N = 10^6$ ,  $\alpha = 0.05$ ,  $\varepsilon = 0.01$

Efektivita metody řídicích veličin závisí na korelačním koeficientu  $\rho_{Y,C}$ , výběrový korelační koeficient  $r_{Y,C}$  má v tomto případě hodnotu

$$r_{Y,C} = 0.778.$$

Závislost náhodných veličin  $Y, C$  znázorňuje graf na obrázku 4.2. Jednotlivé body tohoto grafu mají souřadnice  $[Y_k, C_k]$ , kde  $k = 1, \dots, 10^4$ .



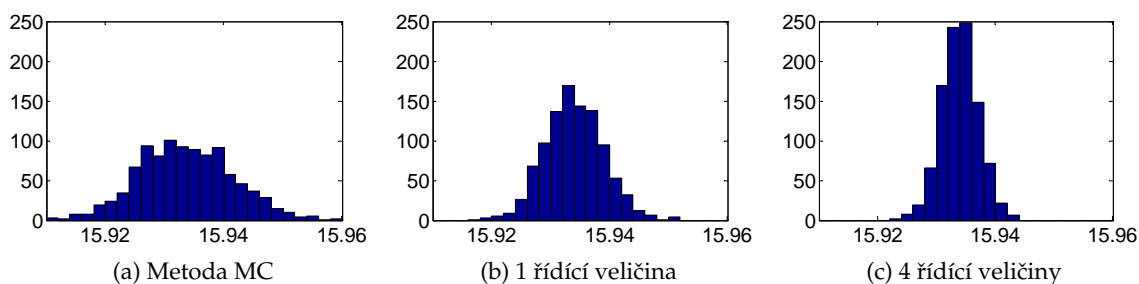
Obrázek 4.2: Závislost náhodných veličin  $Y$  a  $C$ , náhodný výběr o rozsahu  $10^4$

Další možností je použít více řídicích veličin, viz poznámka 4.9. Zde se nabízí jako další řídicí veličiny využít délky ostatních cest z bodu  $A$  do bodu  $B$ , tedy

$$C = (X_1 + X_2 + X_6, X_3 + X_4 + X_6, X_3 + X_5, X_7)$$

a  $\ell^{(c)} = (9, 13, 8, 7)$ . I v tomto případě bylo provedeno  $N = 10^6$  náhodných pokusů, výsledky simulace jsou uvedeny v posledním řádku tabulky 4.2. Rozptyl tohoto estimátoru je přibližně 6krát nižší než rozptyl MC estimátoru.

K porovnání metody MC a obou variant metody řídicích veličin byla celá simulace o rozsahu  $N = 10^6$  zopakována 1000krát, výsledné odhady hodnoty  $\ell$  ukazují histogramy na obrázku 4.3.



Obrázek 4.3: Histogramy odhadů hodnoty  $\ell$  získaných z  $10^3$  simulací

△

Metoda řídicích veličin bude nyní aplikována na data získaná simulací simulací DEDS. Stejná simulace byla v předchozí kapitole vyhodnocována pomocí metody MC.

**Příklad 4.11** (Systém hromadné obsluhy, část 3). Jedná se o systém hromadné obsluhy z příkladu 3.5, parametry úlohy zůstávají stejné. Byly použity tři řídicí veličiny, a to průměry všech náhodných čísel, která byla během jednoho běhu simulace vygenerována postupně z rozdělení  $\text{Exp}(1)$ ,  $\text{Weib}\left(3 \cdot (\Gamma(1.5))^{-1}, 2\right)$  a  $\text{Weib}\left(0.5 \cdot (\Gamma(1.5))^{-1}, 2\right)$ . Díky zvolenému způsobu ukončení simulace byl v každém pokusu vygenerován stejný počet náhodných čísel. Pro střední hodnoty těchto řídicích veličin platí  $\ell^{(c)} = (1, 3, 0.5)$ .

S použitím těchto řídicích veličin byla průměrná doba, kterou zákazník stráví na čerpací stanici, odhadnuta hodnotou  $\hat{\ell} = 3.910$ . Pro výběrový rozptyl odhadu platí

$$s^2 = 9.458 \cdot 10^{-3},$$

oproti MC estimátoru byl tedy rozptyl snížen přibližně 3.6krát. Odhadem 95% intervalu spolehlivosti pro parametr  $\ell$  je interval  $(3.908, 3.912)$ .  $\triangle$

### 4.3 Podmíněná metoda Monte Carlo

Tato metoda redukce rozptylu bývá v literatuře, např [2], označována také jako Rao-Blackwellizace, je založena na poznatku

$$\mathbb{E}[\mathbb{E}(Y|\mathbf{Z})] = \mathbb{E}(Y), \quad (4.4)$$

kde  $Y$  je náhodná veličina a  $\mathbf{Z}$  náhodný vektor, a na následující větě.

**Věta 4.12.** *Nechť  $Y$  je náhodná veličina a  $\mathbf{Z}$  náhodný vektor. Pak*

$$\text{Var}(Y) = \mathbb{E}(\text{Var}(Y|\mathbf{Z})) + \text{Var}(\mathbb{E}(Y|\mathbf{Z})). \quad (4.5)$$

*Důkaz.* Pro důkaz (4.5) stačí dosadit vztahy  $\text{Var}(X) = \mathbb{E}(X^2) - (\mathbb{E}X)^2$  a (4.4), platí

$$\begin{aligned} \mathbb{E}(\text{Var}(Y|\mathbf{Z})) + \text{Var}(\mathbb{E}(Y|\mathbf{Z})) &= \mathbb{E}(\mathbb{E}(Y^2|\mathbf{Z})) - (\mathbb{E}(\mathbb{E}(Y|\mathbf{Z})))^2 = \\ &= \mathbb{E}(Y^2) - (\mathbb{E}Y)^2 = \text{Var}(Y). \end{aligned}$$

□

**Důsledek 4.13.** *Nechť  $Y$  je náhodná veličina a  $\mathbf{Z}$  náhodný vektor, pak*

$$\text{Var}(\mathbb{E}(Y|\mathbf{Z})) \leq \text{Var}(Y).$$

Podmíněná metoda Monte Carlo bude rovněž použita k odhadu hodnoty  $\ell = \mathbb{E}Y$ . Spočívá v nalezení takové náhodné veličiny nebo náhodného vektoru  $\mathbf{Z}$ , že  $\mathbb{E}(Y|\mathbf{Z})$  lze spočítat analyticky. Střední hodnotu  $\mathbb{E}(Y|\mathbf{Z} = \mathbf{z})$  by tedy mělo být možné snadno určit pro libovolnou hodnotu  $\mathbf{z}$ . Z poznatku (4.4) vyplývá, že  $\mathbb{E}(Y|\mathbf{Z})$  je nestranným odhadem  $\ell$ . Podle důsledku 4.13 se navíc nemůže stát, že by odhad získaný pomocí podmíněné metody MC měl vyšší rozptyl než odhad získaný pomocí základní metody MC.

Následuje algoritmus pro řešení obecné úlohy 3.1 pomocí podmíněné metody Monte Carlo.

**Algoritmus 4.14** (Podmíněná metoda MC).

1. Vygenerování náhodného výběru  $\mathbf{Z}_1, \dots, \mathbf{Z}_N$  z pravděpodobnostního rozdělení o hustotě  $g$ .
2. Analytický výpočet  $\mathbb{E}(H(\mathbf{X})|\mathbf{Z}_k)$  pro všechna  $k = 1, \dots, N$ .
3. Odhad hodnoty  $\ell$  výběrovým průměrem z náhodné veličiny  $\mathbb{E}(H(\mathbf{X})|\mathbf{Z})$ , tedy nestranným odhadem

$$\hat{\ell}_c = \frac{1}{N} \sum_{k=1}^N \mathbb{E}(H(\mathbf{X})|\mathbf{Z}_k).$$

4. Odhad  $(1 - \alpha) 100\%$  intervalu spolehlivosti pro parametr  $\ell$ .

Odhadem  $(1 - \alpha) 100\%$  intervalu spolehlivosti pro parametr  $\ell$  je interval

$$\left( \hat{\ell}_c - z_{1-\frac{\alpha}{2}} \frac{s}{\sqrt{N}}, \hat{\ell}_c + z_{1-\frac{\alpha}{2}} \frac{s}{\sqrt{N}} \right),$$

výběrovou směrodatnou odchylku  $s$  lze určit podobně jako v případě metody MC pomocí vztahu

$$s = \sqrt{\frac{1}{N-1} \sum_{k=1}^N \left( \mathbb{E}(H(\mathbf{X})|\mathbf{Z}_k) - \hat{\ell}_c \right)^2}.$$

Použití podmíněné metody MC bude nyní ilustrováno na příkladu 3.4.

**Příklad 4.15** (PERT, část 4). Hledáme opět odhad hodnoty

$$\ell = \mathbb{E}(H(\mathbf{X})),$$

kde

$$H(\mathbf{X}) = \max \{X_1 + X_2 + X_6, X_3 + X_4 + X_6, X_3 + X_5, X_7\}.$$

Systémovou funkci lze zapsat rovněž v alternativním tvaru

$$H(\mathbf{X}) = \max \{ \max \{X_1 + X_2 + X_6, X_3 + X_4 + X_6, X_3 + X_5\}, X_7 \},$$

právě tento tvar využijeme ke konstrukci náhodné veličiny  $Z$ , volme tedy

$$Z = \max \{X_1 + X_2 + X_6, X_3 + X_4 + X_6, X_3 + X_5\}. \quad (4.6)$$

Dále je třeba analyticky vypočítat  $\mathbb{E}(H(\mathbf{X})|Z)$ , platí

$$\mathbb{E}(H(\mathbf{X})|Z = z) = \mathbb{E}(\max \{Z, X_7\} | Z = z) = \mathbb{E}(\max \{z, X_7\}).$$

Uvažujme nyní libovolné pevně zvolené  $z \in \mathbb{R}$  a sestavme distribuční funkci  $F$  náhodné veličiny  $\max\{z, X_7\}$ . Platí

$$F(y) = \mathbb{P}(X_7 < y) \cdot \mathbb{P}(z < y) = (1 - e^{-\lambda_7 y}) \cdot \mathbb{P}(z < y).$$

Hledaným analytickým vyjádřením hodnoty  $\mathbb{E}(H(\mathbf{X}) | Z = z)$  je tedy funkce

$$\mathbb{E}(H(\mathbf{X}) | Z = z) = z + \frac{1}{\lambda_7} e^{-\lambda_7 z}. \quad (4.7)$$

Simulace spočívá ve vygenerování náhodného výběru  $\mathbf{X}_1, \dots, \mathbf{X}_N$ , výpočtu hodnot  $Z_1, \dots, Z_N$  pomocí vztahu (4.6) a následně hodnot  $\mathbb{E}(H(\mathbf{X}) | Z_1), \dots, \mathbb{E}(H(\mathbf{X}) | Z_N)$  pomocí vztahu (4.7). V tomto případě bylo provedeno  $N = 10^6$  náhodných pokusů. Odhadem hodnoty  $\ell$  je

$$\widehat{\ell}_c = \frac{1}{N} \sum_{k=1}^N \mathbb{E}(H(\mathbf{X}) | Z_k) \doteq 15.934.$$

Výběrový rozptyl náhodného výběru z  $\mathbb{E}(H(\mathbf{X}) | Z)$  je nyní přibližně 1.34krát nižší než v případě metody MC. Ostatní výsledky jsou uvedeny v tabulce 4.3.

$\widehat{\ell}_c$	$s$	$s^2$	RSD	interval spolehlivosti	$N_\varepsilon$	čas simulace
15.934	6.758	45.667	0.042%	(15.920, 15.947)	1754292	0.359 s

Tabulka 4.3: Výsledky příkladu 4.15 pro  $N = 10^6$ ,  $\alpha = 0.05$ ,  $\varepsilon = 0.01$

△

#### 4.4 Importance sampling

Často používanou technikou redukce rozptylu je tzv. *importance sampling* (dále IS). V příkladech na konci této sekce bude ukázáno, že pomocí IS lze dosáhnout značné redukce rozptylu zejména při odhadu pravděpodobnosti řídkých jevů.

Metoda IS spočívá ve zvolení nového rozdělení pravděpodobnosti, z něhož jsou generovány vzorky. Toto nové rozdělení by mělo upřednostňovat tzv. důležité vzorky. Například v případě řešení pohotovosti vysoce spolehlivých systémů je při generování vzorků z původního pravděpodobnostního rozdělení dosaženo chybového stavu pouze ve velmi nízkém procentu případů, pro odhad spolehlivosti s dostatečnou přesností je tedy třeba provést mnoho pokusů. V tomto případě jsou „důležitými vzorky“, které se metoda snaží upřednostnit, právě ty stavové vektory, pro něž je dosaženo chybového stavu systému.

Pomocí této metody bude opět řešena obecná úloha 3.1. Úkolem je tedy nalézt

$$\ell = \mathbb{E}_f(H(\mathbf{X})) = \int H(\mathbf{x}) f(\mathbf{x}) \mu(d\mathbf{x}),$$

kde  $H$  je výkonnostní funkce. Index  $f$  zde značí střední hodnotu vzhledem k hustotě pravděpodobnosti  $f$  náhodné veličiny  $\mathbf{X}$ .

Metoda pracuje rovněž s hustotou pravděpodobnosti  $g$ , jež musí splňovat podmínku

$$g(\mathbf{x}) = 0 \Rightarrow H(\mathbf{x}) f(\mathbf{x}) = 0. \quad (4.8)$$

S využitím  $g$  lze  $\ell$  zapsat jako

$$\ell = \int H(\mathbf{x}) \frac{f(\mathbf{x})}{g(\mathbf{x})} g(\mathbf{x}) \mu(d\mathbf{x}) = \mathbb{E}_g \left( H(\mathbf{X}) \frac{f(\mathbf{X})}{g(\mathbf{X})} \right).$$

**Věta 4.16** (IS estimátor). *Necht'*

- $f$  je hustota pravděpodobnosti,
- $g$  je hustota pravděpodobnosti splňující (4.8),
- $H$  je výkonnostní funkce,
- $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N$  jsou navzájem nezávislé realizace náhodného vektoru  $\mathbf{X} \sim g$ .

Pak IS estimátor

$$\hat{\ell} = \frac{1}{N} \sum_{i=1}^N H(\mathbf{X}_i) \frac{f(\mathbf{X}_i)}{g(\mathbf{X}_i)} \quad (4.9)$$

je nestranným odhadem  $\ell = \mathbb{E}_f(H(\mathbf{X}))$ .

*Důkaz.* Estimátor  $\hat{\ell}$  je nestranným odhadem  $\ell$ , neboť

$$\mathbb{E}_g(\hat{\ell}) = \mathbb{E}_g \left( \frac{1}{N} \sum_{i=1}^N H(\mathbf{X}_i) \frac{f(\mathbf{X}_i)}{g(\mathbf{X}_i)} \right) = \frac{1}{N} \sum_{i=1}^N \mathbb{E}_g \left( H(\mathbf{X}) \frac{f(\mathbf{X})}{g(\mathbf{X})} \right) = \ell.$$

□

Hustotě pravděpodobnosti  $g$  se říká IS hustota, rozdělení pravděpodobnosti, které IS hustota definuje, se říká IS rozdělení a poměr obou hustot pravděpodobnosti

$$W(\mathbf{x}) = \frac{f(\mathbf{x})}{g(\mathbf{x})}$$

se nazývá *likelihood ratio* (dále LR). Pokud  $f = g$ , platí  $W \equiv 1$  a odhad (4.9) přejde do tvaru (3.2), jedná se tedy o MC estimátor.

Algoritmus metody IS lze shrnout do následujících kroků.

---

**Algoritmus 4.17** (Metoda IS).

1. Vygenerování náhodného výběru  $\mathbf{X}_1, \dots, \mathbf{X}_N$  z rozdělení pravděpodobnosti o hustotě  $g$ , která splňuje podmínku (4.8).

2. Výpočet hodnot  $Y_1 = H(\mathbf{X}_1) \frac{f(\mathbf{X}_1)}{g(\mathbf{X}_1)}, \dots, Y_N = H(\mathbf{X}_N) \frac{f(\mathbf{X}_N)}{g(\mathbf{X}_N)}$ .

3. Odhad hodnoty  $\ell$  výběrovým průměrem z náhodné veličiny

$$Y = H(\mathbf{X}) \frac{f(\mathbf{X})}{g(\mathbf{X})},$$

tedy nestranným odhadem (4.9).

4. Odhad  $(1 - \alpha) 100\%$  intervalu spolehlivosti pro parametr  $\ell$  jako

$$\left( \hat{\ell} - z_{1-\frac{\alpha}{2}} \frac{s}{\sqrt{N}}, \hat{\ell} + z_{1-\frac{\alpha}{2}} \frac{s}{\sqrt{N}} \right),$$

kde  $s$  je výběrová směrodatná odchylka náhodného výběru  $Y_1, \dots, Y_N$ .

*Poznámka 4.18* (Náhodný vektor s nezávislými složkami). Jestliže jsou jednotlivé složky náhodného vektoru  $\mathbf{X}$  nezávislé, platí pro jeho sdruženou hustotu pravděpodobnosti

$$f(\mathbf{x}) = \prod_{j=1}^n f_j(x_j), \quad g(\mathbf{x}) = \prod_{j=1}^n g_j(x_j)$$

a vzorec (4.9) pro odhad střední hodnoty  $\ell$  přejde do tvaru

$$\hat{\ell} = \frac{1}{N} \sum_{i=1}^N H(\mathbf{X}_i) \prod_{j \in I} \frac{f_j(X_{ij})}{g_j(X_{ij})},$$

kde  $I = \{j \in \{1, \dots, n\} : f_j \neq g_j\}$  a  $X_{ij}$  je  $j$ -tá složka vektoru  $\mathbf{X}_i$ .

Při praktickém použití metody IS bývá množina  $I$  (tzv. množina důležitých komponent) často ostrou podmnožinou množiny  $\{1, 2, \dots, n\}$ , do LR jsou tak zahrnuty pouze některé složky náhodného vektoru  $\mathbf{X}$ .

Dále budou v této práci používány pouze náhodné vektory, jejichž jednotlivé složky jsou nezávislé. Vícerozměrným rozdělením pravděpodobnosti tedy bude vždy myšleno rozdělení s nezávislými složkami.

Nyní nastává důležitá otázka, jak zvolit hustotu pravděpodobnosti  $g$ . Na rozdíl od podmíněné metody MC, jejíž použití vede oproti základní metodě MC vždy k estimátoru s nižším nebo stejným rozptylem, může v případě metody IS nevhodná volba  $g$  zvýšit rozptyl estimátoru a snížit tak efektivitu simulace.

#### 4.4.1 Optimální hustota pravděpodobnosti

Hustota pravděpodobnosti  $g$  by měla být zvolena tak, aby rozptyl náhodné veličiny  $Y = H(\mathbf{X}) W(\mathbf{X})$  byl minimální.

**Věta 4.19.** *Nechť  $f$  je hustota pravděpodobnosti náhodného vektoru  $\mathbf{X}$ ,  $H$  je výkonnostní funkce a  $g$  je IS hustota splňující podmínku (4.8).*

*Pak*

$$\mathrm{Var}_g \left( H(\mathbf{X}) \frac{f(\mathbf{X})}{g(\mathbf{X})} \right) \quad (4.10)$$

*nabývá svého minima pro funkci*

$$\arg\min_g \mathrm{Var}_g \left( H(\mathbf{X}) \frac{f(\mathbf{X})}{g(\mathbf{X})} \right) = \frac{|H(\mathbf{x})| f(\mathbf{x})}{\int |H(\mathbf{x})| f(\mathbf{x}) \mu(d\mathbf{x})} \stackrel{\text{ozn.}}{=} g^*(\mathbf{x}). \quad (4.11)$$

*Důkaz.* Pro důkaz věty bude použit vztah  $\mathrm{Var} X = \mathbb{E}(X^2) - (\mathbb{E}X)^2$  a z něj plynoucí nerovnost  $\mathbb{E}(X^2) \geq (\mathbb{E}X)^2$ . Pro všechny přípustné funkce  $g$  platí

$$\begin{aligned} \mathrm{Var}_g \left( H(\mathbf{X}) \frac{f(\mathbf{X})}{g(\mathbf{X})} \right) &= \mathbb{E}_g \left[ \left( H(\mathbf{X}) \frac{f(\mathbf{X})}{g(\mathbf{X})} \right)^2 \right] - \left[ \mathbb{E}_g \left( H(\mathbf{X}) \frac{f(\mathbf{X})}{g(\mathbf{X})} \right) \right]^2 = \\ &= \mathbb{E}_g \left[ \left( |H(\mathbf{X})| \frac{f(\mathbf{X})}{g(\mathbf{X})} \right)^2 \right] - \left[ \int H(\mathbf{x}) f(\mathbf{x}) \mu(d\mathbf{x}) \right]^2. \end{aligned}$$

Výraz  $\left[ \int H(\mathbf{x}) f(\mathbf{x}) \mu(d\mathbf{x}) \right]^2$  nezávisí na funkci  $g$ , minimalizace výrazu (4.10) tedy odpovídá minimalizaci výrazu

$$\mathbb{E}_g \left[ \left( |H(\mathbf{X})| \frac{f(\mathbf{X})}{g(\mathbf{X})} \right)^2 \right]. \quad (4.12)$$

Pro (4.12) platí

$$\begin{aligned} \mathbb{E}_g \left[ \left( |H(\mathbf{X})| \frac{f(\mathbf{X})}{g(\mathbf{X})} \right)^2 \right] &\geq \left[ \mathbb{E}_g \left( |H(\mathbf{X})| \frac{f(\mathbf{X})}{g(\mathbf{X})} \right) \right]^2 = \\ &= \left[ \int |H(\mathbf{x})| f(\mathbf{x}) \mu(d\mathbf{x}) \right]^2 = \int |H(\mathbf{x})| f(\mathbf{x}) \int |H(\mathbf{y})| f(\mathbf{y}) \mu(d\mathbf{y}) \mu(d\mathbf{x}) = \\ &= \int H^2(\mathbf{x}) f^2(\mathbf{x}) \frac{\int |H(\mathbf{y})| f(\mathbf{y}) \mu(d\mathbf{y})}{|H(\mathbf{x})| f(\mathbf{x})} \mu(d\mathbf{x}) = \int H^2(\mathbf{x}) \frac{f^2(\mathbf{x})}{\frac{|H(\mathbf{x})| f(\mathbf{x})}{\int |H(\mathbf{y})| f(\mathbf{y}) \mu(d\mathbf{y})}} \mu(d\mathbf{x}). \end{aligned}$$

Jelikož pro všechny přípustné funkce  $g$  platí

$$\int H^2(\mathbf{x}) \frac{f^2(\mathbf{x})}{g(\mathbf{x})} \mu(d\mathbf{x}) = \mathbb{E}_g \left[ \left( |H(\mathbf{X})| \frac{f(\mathbf{X})}{g(\mathbf{X})} \right)^2 \right] \geq \int H^2(\mathbf{x}) \frac{f^2(\mathbf{x})}{\frac{|H(\mathbf{x})| f(\mathbf{x})}{\int |H(\mathbf{y})| f(\mathbf{y}) \mu(d\mathbf{y})}} \mu(d\mathbf{x}),$$

výraz (4.10) zřejmě nabývá minima pro funkci  $g^*(\mathbf{x})$ . □



**Důsledek 4.20.** Pokud  $H(\mathbf{x}) \geq 0$ , zřejmě platí

$$g^*(\mathbf{x}) = \frac{H(\mathbf{x}) f(\mathbf{x})}{\ell} \quad (4.13)$$

a navíc

$$\text{Var}_{g^*}(\hat{\ell}) = \text{Var}_{g^*}(H(\mathbf{X}) W(\mathbf{X})) = \text{Var}_{g^*}(\ell) = 0.$$

Hustota pravděpodobnosti (4.11) se nazývá optimální IS hustotou. Jedná se ovšem pouze o teoretickou optimální funkci, jelikož závisí na hledané hodnotě  $\ell$ , není možné ji vyjádřit předem.

*Poznámka 4.21* (Degenerace LR, viz [1]). Častým problémem, který se objevuje při redukci rozptylu pomocí metody IS, je tzv. degenerace LR. Nastává zejména při řešení úloh vyšší dimenze. Je-li pro některé realizace  $\mathbf{X}_i$  náhodného vektoru  $\mathbf{X}$  hodnota  $f(\mathbf{X}_i)$  nezanedbatelná a hodnota  $g(\mathbf{X}_i)$  blízká nule, je pro tyto vektory  $\mathbf{X}_i$  poměr  $W(\mathbf{X}_i)$  vysoký. Naopak pro jiné realizace  $\mathbf{X}_i$  může být hodnota  $f(\mathbf{X}_i)$  blízká nule a hodnota  $g(\mathbf{X}_i)$  vysoká, jejich poměr  $W(\mathbf{X}_i)$  je pak blízký nule. Při takové volbě IS rozdělení má náhodná veličina  $W(\mathbf{X})$  vysoký rozptyl vzhledem k  $g$ .

#### 4.4.2 Metoda minimalizace rozptylu

Metoda minimalizace rozptylu, dále VM (z anglického *variance minimization*), je metodou výběru IS hustoty  $g$ . Je obvyklé volit hustotu pravděpodobnosti  $g$  ze stejné pravděpodobnostní rodiny, z níž pochází hustota pravděpodobnosti  $f$ .

Nechť hustota pravděpodobnosti  $f$  závisí na parametru (či vektoru parametrů)  $\theta$ , lze ji tedy značit  $f(\cdot; \theta)$ . Nová hustota pravděpodobnosti bude hledána mezi funkcemi z množiny

$$\{f(\cdot; \eta), \eta \in \Theta\},$$

které splňují podmínku

$$f(\mathbf{x}; \eta) = 0 \Leftrightarrow f(\mathbf{x}; \theta) = 0. \quad (4.14)$$

Problém hledání funkce  $g$  tak přechází v řešení minimalizační úlohy

$$\min_{\eta \in \Theta} \text{Var}_{\eta}(H(\mathbf{X}) W(\mathbf{X}; \theta, \eta)),$$

kde

$$W(\mathbf{X}; \theta, \eta) = \frac{f(\mathbf{X}; \theta)}{f(\mathbf{X}; \eta)},$$

$\text{Var}_{\eta}$  značí rozptyl vzhledem k hustotě  $f(\mathbf{x}; \eta)$  a  $\mathbb{E}_{\eta}$  značí střední hodnotu vzhledem k hustotě  $f(\mathbf{x}; \eta)$ . Podobně jako v důkazu věty 4.19 lze odvodit, že

$$\min_{\eta \in \Theta} \text{Var}_{\eta}(H(\mathbf{X}) W(\mathbf{X}; \theta, \eta)) = \min_{\eta \in \Theta} \mathbb{E}_{\theta}(H^2(\mathbf{X}) W(\mathbf{X}; \theta, \eta)).$$

Hledanou hodnotu

$$\eta^* = \underset{\eta \in \Theta}{\operatorname{argmin}} \mathbb{E}_{\theta}(H^2(\mathbf{X}) W(\mathbf{X}; \theta, \eta)) \quad (4.15)$$

je možné odhadnout simulačně pomocí nahrazení střední hodnoty příslušným výběrovým průměrem.

Jelikož pro vektor parametrů  $\zeta \in \Theta$  platí

$$\mathbb{E}_{\theta} \left( H^2(\mathbf{X}) \frac{f(\mathbf{X}; \theta)}{f(\mathbf{X}; \eta)} \right) = \mathbb{E}_{\zeta} \left( H^2(\mathbf{X}) \frac{f(\mathbf{X}; \theta)}{f(\mathbf{X}; \eta)} \frac{f(\mathbf{X}; \theta)}{f(\mathbf{X}; \zeta)} \right),$$

lze hodnotu  $\eta^*$  lze zapsat rovněž vztahem

$$\eta^* = \operatorname{argmin}_{\eta \in \Theta} \mathbb{E}_{\zeta} \left( H^2(\mathbf{X}) W(\mathbf{X}; \theta, \eta) W(\mathbf{X}; \theta, \zeta) \right).$$

Díky tomuto zápisu lze proces minimalizace rozptylu formulovat jako iterační metodu.

---

**Algoritmus 4.22** (Iterační metoda minimalizace rozptylu).

1. Zvolit počáteční vektor parametrů  $\eta^{(0)}$ , například  $\eta^{(0)} = \theta$ , nastavit  $k = 1$ .
2. Vygenerovat náhodný výběr  $\mathbf{X}_1, \dots, \mathbf{X}_N$  z rozdělení pravděpodobnosti o hustotě  $f(\cdot; \eta^{(0)})$ .
3. Určit

$$\eta^{(k)} = \operatorname{argmin}_{\eta \in \Theta} \frac{1}{N} \sum_{k=1}^N \left( H^2(\mathbf{X}_k) W(\mathbf{X}_k; \theta, \eta) W(\mathbf{X}_k; \theta, \eta^{(k-1)}) \right) \quad (4.16)$$

například pomocí vhodných numerických metod.

4. Není-li splněna ukončující podmínka, navýšit iterační proměnnou  $k$  o 1 a pokračovat krokem 2.
  5. Aproximovat optimální vektor  $\eta^*$  vektorem  $\eta^{(k)}$  a použít hustotu pravděpodobnosti  $f(\mathbf{X}; \eta^{(k)})$  jako IS hustotu při řešení úlohy 3.1 pomocí algoritmu 4.17.
- 

Ukončující podmínkou metod pro odhad parametrů IS hustoty může být například dosažení předem stanoveného počtu kroků nebo stagnace metody na určitém vektoru parametrů.

#### 4.4.3 Metoda Cross-Entropy

Metoda Cross-Entropy, dále CE, je další možností, jak nalézt IS hustotu, pro níž bude rozptyl estimátoru (4.9) co nejmenší. Opět předpokládáme, že původní hustota pravděpodobnosti má tvar  $f(\cdot; \theta)$  a IS hustota patří do množiny

$$\{f(\cdot; \eta), \eta \in \Theta\}.$$

Tato metoda je založena na minimalizaci Kullback-Leiblerovy divergence hledané IS hustoty  $f(\cdot; \boldsymbol{\eta})$  a optimální IS hustoty  $g^*$  vzhledem k vektoru parametrů  $\boldsymbol{\eta} \in \Theta$ . Je tedy řešena minimalizační úloha

$$\min_{\boldsymbol{\eta} \in \Theta} \mathcal{D}(g^*, f(\cdot; \boldsymbol{\eta})). \quad (4.17)$$

Pro Kullback-Leiblerovu divergenci hustot pravděpodobnosti  $g, f$  platí

$$\begin{aligned} \mathcal{D}(g, f) &= \mathbb{E}_g \left[ \ln \frac{g(\mathbf{X})}{f(\mathbf{X})} \right] = \int g(\mathbf{x}) \ln \frac{g(\mathbf{x})}{f(\mathbf{x})} \mu(d\mathbf{x}) = \\ &= \int g(\mathbf{x}) \ln g(\mathbf{x}) \mu(d\mathbf{x}) - \int g(\mathbf{x}) \ln f(\mathbf{x}) \mu(d\mathbf{x}), \end{aligned}$$

kde výraz  $\int g(\mathbf{x}) \ln g(\mathbf{x}) \mu(d\mathbf{x})$  nezávisí na  $f$ . Za dodatečného předpokladu  $H(\mathbf{x}) \geq 0$ , díky němuž lze dosadit rovnost 4.13, je tedy minimalizační úloha (4.17) ekvivalentní úloze maximalizace

$$\max_{\boldsymbol{\eta} \in \Theta} \int H(\mathbf{x}) f(\mathbf{x}; \boldsymbol{\eta}) \ln f(\mathbf{x}; \boldsymbol{\eta}) \mu(d\mathbf{x}) = \max_{\boldsymbol{\eta} \in \Theta} \mathbb{E}_{\boldsymbol{\eta}} (H(\mathbf{X}) \ln f(\mathbf{X}; \boldsymbol{\eta})). \quad (4.18)$$

Podobně jako v případě metody VM lze hledanou hodnotu

$$\boldsymbol{\eta}^* = \operatorname{argmax}_{\boldsymbol{\eta} \in \Theta} \mathbb{E}_{\boldsymbol{\eta}} (H(\mathbf{X}) \ln f(\mathbf{X}; \boldsymbol{\eta}))$$

odhadnout pomocí metody MC a rovněž existuje iterační varianta metody CE.

Úloha (4.18) je zřejmě ekvivalentní s úlohou

$$\max_{\boldsymbol{\eta} \in \Theta} \mathbb{E}_{\boldsymbol{\zeta}} (H(\mathbf{X}) W(\mathbf{X}; \boldsymbol{\theta}, \boldsymbol{\zeta}) \ln f(\mathbf{X}; \boldsymbol{\eta})),$$

s využitím tohoto zápisu lze proces odhadu vektoru  $\boldsymbol{\eta}^*$  chápat jako iterační metodu, viz následující algoritmus. [8, str. 69]

**Algoritmus 4.23** (Metoda CE). Vstupem algoritmu je původní vektor parametrů  $\boldsymbol{\theta}$ .

1. Zvolit počáteční vektor parametrů  $\boldsymbol{\eta}^{(0)}$ , například  $\boldsymbol{\eta}^{(0)} = \boldsymbol{\theta}$ . Nastavit  $k = 1$ .
2. Vygenerovat náhodný výběr  $\mathbf{X}_1, \dots, \mathbf{X}_N$  z rozdělení pravděpodobnosti o hustotě  $f(\cdot; \boldsymbol{\eta}^{(k-1)})$ .

3. Určit

$$\boldsymbol{\eta}^{(k)} = \operatorname{argmax}_{\boldsymbol{\eta}} \frac{1}{N} \sum_{i=1}^N H(\mathbf{X}_i) W(\mathbf{X}_i; \boldsymbol{\theta}, \boldsymbol{\eta}^{(k-1)}) \ln f(\mathbf{X}_i; \boldsymbol{\eta}). \quad (4.19)$$

4. Není-li splněna ukončující podmínka, navýšit iterační proměnnou  $k$  o 1 a pokračovat krokem 2.
5. Použít hustotu pravděpodobnosti  $f(\mathbf{X}; \boldsymbol{\eta}^{(k)})$  jako IS hustotu při řešení úlohy 3.1 pomocí algoritmu 4.17.

Vzorec (4.19) pro aktualizaci vektoru  $\boldsymbol{\eta}^{(k-1)}$  na vektor  $\boldsymbol{\eta}^{(k)}$  lze pro většinu běžně používaných rozdělení pravděpodobnosti značně zjednodušit, což je výhodou oproti metodě VM. Jestliže jsou jednotlivé složky náhodného vektoru  $\mathbf{X}$  nezávislé, jejich rozdělení pravděpodobnosti patří do exponenciální rodiny (zkráceně NEF z anglického *natural exponential family*) a vektor  $\boldsymbol{\eta}^{(k-1)}$  je střední hodnotou příslušného sdruženého rozdělení pravděpodobnosti, platí pro jednotlivé složky  $\eta_j^{(k)}$  vektoru  $\boldsymbol{\eta}^{(k)}$

$$\eta_j^{(k)} = \frac{\sum_{i=1}^N H(\mathbf{X}_i) W(\mathbf{X}_i; \boldsymbol{\theta}, \boldsymbol{\eta}^{(k-1)}) X_{ij}}{\sum_{i=1}^N H(\mathbf{X}_i) W(\mathbf{X}_i; \boldsymbol{\theta}, \boldsymbol{\eta}^{(k-1)})}, \quad (4.20)$$

kde  $X_{ij}$  je  $j$ -tá složka vektoru  $\mathbf{X}_i$ , viz [8]. Jedná se například o (vícerozměrné) exponenciální rozdělení parametrizované střední hodnotou nebo (vícerozměrné) Bernoulliho rozdělení.

Podobně v případě, že náhodný vektor  $\mathbf{X}$  pochází z (vícerozměrného) Weibullova rozdělení s pevným parametrem tvaru  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$  a parametrem měřítka

$$\boldsymbol{\lambda} = \left( \left( \eta_1^{(k-1)} \right)^{-1}, \dots, \left( \eta_n^{(k-1)} \right)^{-1} \right),$$

lze opět vektor  $\boldsymbol{\eta}^{(k-1)}$  aktualizovat po složkách, platí

$$\eta_j^{(k)} = \left( \frac{\sum_{i=1}^N H(\mathbf{X}_i) W(\mathbf{X}_i; \boldsymbol{\theta}, \boldsymbol{\eta}^{(k-1)}) X_{ij}^{\alpha_j}}{\sum_{i=1}^N H(\mathbf{X}_i) W(\mathbf{X}_i; \boldsymbol{\theta}, \boldsymbol{\eta}^{(k-1)})} \right)^{\frac{1}{\alpha_j}}. \quad (4.21)$$

*Poznámka 4.24* (Vyhlašovací parametr, viz [8, str. 134]). Krok 3 algoritmu 4.23 je možné nahradit krokem

- Vypočít

$$\tilde{\boldsymbol{\eta}}^{(k)} = \underset{\boldsymbol{\eta}}{\operatorname{argmax}} \frac{1}{N} \sum_{i=1}^N H(\mathbf{X}_i) W(\mathbf{X}_i; \boldsymbol{\theta}, \boldsymbol{\eta}^{(k-1)}) \ln f(\mathbf{X}_i; \boldsymbol{\eta})$$

a určit  $\boldsymbol{\eta}^{(k)}$  jako

$$\boldsymbol{\eta}^{(k)} = \alpha \tilde{\boldsymbol{\eta}}^{(k)} + (1 - \alpha) \boldsymbol{\eta}^{(k-1)},$$

kde  $\alpha \in (0, 1)$  je předem zvolený vyhlašovací parametr.

K řešení příkladu 3.4 bude nyní použita metoda IS s odhadem parametrů IS hustoty pomocí metod CE a VM.

**Příklad 4.25** (PERT, část 5). Náhodný vektor  $\mathbf{X}$  pochází z vícerozměrného exponenciálního rozdělení. Jelikož bude aplikována metoda CE, je vhodné použít parametrizaci pomocí střední hodnoty, hustotou pravděpodobnosti náhodného vektoru  $\mathbf{X}$  je tedy funkce

$$f(\mathbf{x}; \boldsymbol{\theta}) = \exp \left( - \sum_{i=1}^7 \frac{x_i}{\theta_i} \right) \prod_{i=1}^7 \frac{1}{\theta_i},$$

kde  $\theta = (1, 2, \dots, 7)$ . Prvním úkolem je tedy nalézt pomocí metody CE vhodnou IS hustotu ve tvaru  $f(x; \eta)$ . S použitím algoritmu 4.23 a vzorce (4.20) byl získán vektor

$$\eta = (1.019, 2.062, 3.459, 4.701, 5.624, 7.848, 8.368).$$

Bylo provedeno celkem 10 iterací metody CE, v každé iteraci byl vygenerován náhodný vektor o rozsahu  $10^4$ , odhad parametrů trval celkem 0.063 s. Pomocí metody minimalizace rozptylu byl získán přibližně stejný vektor parametrů

$$\tilde{\eta} = (1.007, 2.085, 3.404, 4.781, 5.589, 7.713, 8.350),$$

obecně však metody CE a VM nemusí dojít ke stejnému řešení. Odhad parametrů pomocí metody VM trval 11.5 s, bylo provedeno 10 iterací a v každé z nich  $10^3$  náhodných pokusů, pro řešení minimalizační úlohy 4.16 byla použita vestavěná funkce `fminsearch` programu Matlab.

Druhým krokem řešení úlohy je aplikace algoritmu 4.17 se získanou IS hustotou  $f(x; \eta)$ . Bylo vykonáno  $N = 10^6$  náhodných pokusů, pro všechna  $k \in \{1, \dots, N\}$  byl nejprve vygenerován vektor  $\mathbf{X}_k$  z vícerozměrného exponenciálního rozdělení o hustotě  $f(x; \eta)$  a následně vypočteny hodnoty  $Y_k = H(\mathbf{X}_k) W(\mathbf{X}_k; \theta, \eta)$ , kde  $W(\mathbf{X}_k; \theta, \eta) = (f(\mathbf{X}_k; \eta))^{-1} f(\mathbf{X}_k; \theta)$ . Nestranným odhadem hodnoty  $\ell$  je výběrový průměr

$$\hat{\ell} = \frac{1}{N} \sum_{i=k}^N H(\mathbf{X}_k) W(\mathbf{X}_k; \theta, \eta) \doteq 15.938.$$

Další výsledky jsou uvedeny v tabulce 4.4. Výběrový rozptyl  $s^2$  je přibližně pětikrát nižší než v případě metody MC, pro dosažení stejné přesnosti odhadu tedy stačí vykonat pětikrát méně náhodných pokusů.

$\hat{\ell}$	$s$	$s^2$	RSD	interval spolehlivosti	$N_\varepsilon$
15.938	3.479	12.105	$2.18 \cdot 10^{-2}\%$	(15.931, 15.944)	464989

Tabulka 4.4: Výsledky příkladu 4.25 pro  $N = 10^6$ ,  $\alpha = 0.05$ ,  $\varepsilon = 0.01$

△

#### 4.4.4 Metoda CE pro kvantifikaci řídkých jevů

Řešme nyní metodou IS s odhadem parametrů rozdělení pomocí CE obecnou úlohu 3.7, která pro vhodnou volbu parametru  $\gamma$  vede na kvantifikaci řídkých jevů.

Pro odhad parametrů by bylo možné použít základní metodu CE, nicméně v případě nízké hodnoty  $\ell$  by ve druhém kroku metody mohlo dojít k vygenerování pouze takových realizací  $\mathbf{X}_i$  náhodné veličiny  $\mathbf{X}$ , pro které  $H(\mathbf{X}_i) = 0$ , a nebylo by tak možné vypočíst nový vektor parametrů rozdělení. Tento problém lze částečně vyřešit přidáním kroku

- Pokud  $H(\mathbf{X}_1) = \dots = H(\mathbf{X}_N) = 0$ , opakovat krok 2.

mezi 2. a 3. krok algoritmu. Generování náhodného výběru by tak bylo opakováno, dokud by pro některý ze vzorků neplatilo  $H(\mathbf{X}_i) = 1$ . Tento postup ale často způsobuje degeneraci LR. Existuje však modifikace metody CE, která tento problém řeší efektivně, spočívá v zápisu funkce  $H(\mathbf{X})$  jako funkce proměnné  $\gamma$ , tedy

$$H(\mathbf{X}) = H(\mathbf{X}, \gamma) = \mathbb{I}_{S(\mathbf{X} \geq \gamma)},$$

a použití následujícího algoritmu, viz [1, str. 238].

---

**Algoritmus 4.26** (Metoda CE pro kvantifikaci řídkých jevů).

1. Zvolit počáteční vektor parametrů  $\boldsymbol{\eta}^{(0)} = \boldsymbol{\theta}$ , nastavit  $k = 1$ .
2. Vygenerovat náhodný výběr  $\mathbf{X}_1, \dots, \mathbf{X}_N$  z rozdělení pravděpodobnosti o hustotě  $f(\cdot, \boldsymbol{\eta}^{(k-1)})$ .
3. Vypočítat hodnoty  $S(\mathbf{X}_1), \dots, S(\mathbf{X}_N)$  a seřadit je do posloupnosti  $S_{(1)} \leq \dots \leq S_{(N)}$ . Určit výběrový  $(1 - \rho)$ -kvantil jako

$$\hat{\gamma}_k = S_{(\lceil (1-\rho)N \rceil)}.$$

Pokud  $\hat{\gamma}_k > \gamma$ , nastavit  $\hat{\gamma}_k = \gamma$ .

4. Určit

$$\boldsymbol{\eta}^{(k)} = \underset{\boldsymbol{\eta}}{\operatorname{argmax}} \frac{1}{N} \sum_{i=1}^N \mathbb{I}_{\{S(\mathbf{X}_i) \geq \hat{\gamma}_k\}} W(\mathbf{X}_i; \boldsymbol{\theta}, \boldsymbol{\eta}^{(k-1)}) \ln f(\mathbf{X}_i; \boldsymbol{\eta}).$$

5. Jestliže  $\hat{\gamma}_k < \gamma$ , navýšit iterační proměnnou  $k$  o 1 a pokračovat krokem 2.
  6. Použít hustotu pravděpodobnosti  $f(\mathbf{X}; \boldsymbol{\eta}^{(k)})$  jako IS hustotu při řešení úlohy 3.1 pomocí algoritmu 4.17.
- 

Analogicky lze modifikovat také metodu VM a uzpůsobit ji tak pro kvantifikaci řídkých jevů.

S použitím této varianty metod CE a VM bude nyní odhadnuta pravděpodobnost řídkého jevu definovaného v příkladu 3.6, kde byla tato pravděpodobnost hledána pomocí metody MC.

**Příklad 4.27** (Nejkratší cesta, část 3). Uvažujme opět systém znázorněný na obrázku 2.5 a předpokládejme, že náhodný vektor  $\mathbf{X}$  pochází z Weibullova rozdělení s parametrem tvaru  $\alpha = 2$  a parametrem měřítka

$$\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_6) = (2^{-1}, 4^{-1}, 6^{-1}, 2^{-1}, 4^{-1}, 6^{-1}).$$

Úkolem je určit pomocí metody IS střední hodnotu náhodné veličiny

$$H(\mathbf{X}) = I_{\{S(\mathbf{X}) \geq \gamma\}},$$

kde  $S(\mathbf{X})$  je výkonnostní funkce, která udává délku nejkratší cesty z A do B, a  $\gamma = 16$ . Pro odhad parametrů IS hustoty byl použit nejprve algoritmus 4.26 s volbou  $\rho = 10^{-1}$  a  $\boldsymbol{\eta}^{(0)} = (2, 4, 6, 2, 4, 6)$ . Jelikož byly optimalizovány parametry Weibullova rozdělení, byl pro jejich aktualizaci použit vzorec (4.21). Do dosažení úrovně  $\gamma$  byly provedeny 4 iterace, přičemž výběrovému  $(1 - \rho)$ -kvantilu odpovídaly postupně hodnoty  $\hat{\gamma}_1 = 8.34$ ,  $\hat{\gamma}_2 = 11.32$ ,  $\hat{\gamma}_3 = 13.77$  a  $\hat{\gamma}_4 = 15.97$ . Po dosažení úrovně  $\gamma = 16$  bylo pro zpřesnění dále provedeno 10 iterací metody CE pomocí algoritmu 4.23. Výsledkem je vektor převrácených hodnot parametrů IS hustoty

$$\boldsymbol{\eta} = (3.203, 9.904, 13.451, 4.223, 4.150, 6.328).$$

Jako IS hustota byla použita hustota Weibullova rozdělení s parametrem tvaru  $\alpha = 2$  a parametrem měřítka  $(\eta_1^{-1}, \dots, \eta_6^{-1})$ . Bylo provedeno  $N = 10^6$  náhodných pokusů, odhadem hodnoty  $\ell$  je  $\hat{\ell} = 5.350 \cdot 10^{-6}$ . Odhadem 95% intervalu pro parametr  $\ell$  je interval

$$(5.254 \cdot 10^{-6}, 5.446 \cdot 10^{-6}).$$

Pro dosažení přesnosti  $\varepsilon = 10^{-7}$  s pravděpodobností 0.95 by stačilo provést  $N_\varepsilon = 9.18 \cdot 10^5$  náhodných pokusů, což je přibližně 2100krát méně než v případě metody MC, viz příklad 3.6.

Parametry byly odhadovány také pomocí metody VM přizpůsobené pro kvantifikaci řídkých jevů, byl získán vektor převrácených hodnot parametrů IS hustoty

$$\boldsymbol{\eta}_{VM} = (3.161, 9.549, 13.002, 4.091, 4.088, 6.005).$$

Použití tohoto vektoru parametrů vede k mírně vyšší redukci rozptylu než použití vektoru  $\boldsymbol{\eta}$ , metoda VM je ale výpočetně náročnější. Tabulka 4.5 porovnává obě varianty metody IS s výsledky získanými pomocí metody MC v příkladu 2.4.

metoda	$N$	$\hat{\ell}$	$s^2$	RSD	$N_\varepsilon$	redukce
MC	$10^9$	$5.31 \cdot 10^{-6}$	$5.31 \cdot 10^{-6}$	1.37%	$2.04 \cdot 10^9$	$1 \times$
IS + CE	$10^6$	$5.35 \cdot 10^{-6}$	$2.39 \cdot 10^{-9}$	0.91%	$9.2 \cdot 10^5$	$2093 \times$
IS + VM	$10^6$	$5.27 \cdot 10^{-6}$	$2.27 \cdot 10^{-9}$	0.90%	$8.7 \cdot 10^5$	$2200 \times$

Tabulka 4.5: Porovnání výsledků metod MC a IS

△

## 5 Analýza citlivosti DE systémů

Analýza citlivosti vyhodnocuje vliv drobných změn vstupních parametrů systému na hodnotu  $\ell = \mathbb{E}(H(\mathbf{X}))$ . Lze tak identifikovat nejvýznamnější komponenty systému. V řeči analýzy pohotovosti vysoce spolehlivých systémů se jedná o identifikaci komponent, které nejčastěji způsobují poruchy systému. Výsledky analýzy citlivosti tak mohou sloužit jako podklad pro zdokonalení systému vedoucí ke zvýšení jeho spolehlivosti. Tato kapitola vychází zejména z [2].

Odhadovaná střední hodnota je chápána jako funkce vektoru vstupních parametrů  $\boldsymbol{\theta} = (u_1, \dots, u_m)$ , tedy  $\ell = \ell(\boldsymbol{\theta})$ . Úkolem analýzy citlivosti je pak odhadnout gradient funkce  $\ell(\boldsymbol{\theta})$ , který indikuje právě míru vlivu jednotlivých parametrů  $\theta_1, \dots, \theta_m$  na výslednou hodnotu  $\ell(\boldsymbol{\theta})$ .

Vektor  $\boldsymbol{\theta}$  může být vektorem parametrů hustoty pravděpodobnosti  $f(\mathbf{X})$   $n$ -rozměrného náhodného vektoru  $\mathbf{X}$  nebo parametrů výkonnostní funkce  $H(\mathbf{X})$ . V obecném případě se jedná o kombinaci obou typů parametrů, tj.  $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ , kde  $\boldsymbol{\theta}_1$  je vektorem parametrů rozdělení a  $\boldsymbol{\theta}_2$  je vektorem parametrů výkonností funkce. V této kapitole tedy bude řešena následující obecná úloha:

**Úloha 5.1.** Nechť  $\mathbf{X} = (X_1, \dots, X_n) \in \mathbb{R}^n$  je náhodný vektor pocházející z rozdělení pravděpodobnosti o hustotě  $f(\mathbf{X}) = f(\mathbf{X}; \boldsymbol{\theta}_1)$  a  $H(\mathbf{X}) = H(\mathbf{X}; \boldsymbol{\theta}_2)$  je výkonnostní funkce. Určete gradient funkce  $\ell(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}_1}(H(\mathbf{X}; \boldsymbol{\theta}_2))$ , kde  $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ .

Odhad střední hodnoty je rovněž chápán jako funkce vektoru  $\boldsymbol{\theta}$ , tedy  $\hat{\ell} = \hat{\ell}(\boldsymbol{\theta})$ .

### 5.1 Metoda centrálních diferencí

Při numerickém odhadu gradientu funkce  $\ell(\boldsymbol{\theta})$  jsou derivace obvykle nahrazovány konečnými diferencemi. Je možné použít dopředné difference a odhadovat  $i$ -tý prvek vektoru  $\nabla \ell(\boldsymbol{\theta})$  hodnotou

$$\frac{\partial \ell(\boldsymbol{\theta})}{\partial \theta_i} \approx \frac{\ell(\boldsymbol{\theta} + \delta \mathbf{e}_i) - \ell(\boldsymbol{\theta})}{\delta},$$

kde  $\delta > 0$  a  $\mathbf{e}_i$  je  $i$ -tý vektor standardní báze prostoru  $\mathbb{R}^m$ , nebo centrální difference a použít tak odhad

$$\frac{\partial \ell(\boldsymbol{\theta})}{\partial \theta_i} \approx \frac{\ell(\boldsymbol{\theta} + \frac{\delta}{2} \mathbf{e}_i) - \ell(\boldsymbol{\theta} - \frac{\delta}{2} \mathbf{e}_i)}{\delta}.$$

Uvedené odhady nejsou nestranné, s ohledem na vychýlenost má však lepší vlastnosti odhad sestavený s využitím centrálních diferencí, v případě analýzy citlivosti je tedy obecně preferován. Hodnota  $\delta$  by měla být dostatečně nízká, aby nedocházelo k výraznému zkreslení výsledku v důsledku vychýlení, nízká hodnota parametru  $\delta$  však zároveň vede k vysokému rozptylu estimátoru, je tedy třeba zvolit vhodný kompromis.

Při implementaci této metody je vhodné využít metodu redukce rozptylu CRN (z anglického *common random numbers*). Metoda CRN slouží k odhadu hodnoty  $\mathbb{E}(Y) - \mathbb{E}(Z)$ , kde  $Y$  a  $Z$  jsou kladně korelované náhodné veličiny. Právě korelace náhodných veličin  $Y$  a  $Z$  určuje míru redukce rozptylu odhadu. Více o CRN lze nalézt například v [7].



Parametr  $\theta_i$  může být parametrem rozdělení náhodného vektoru  $\mathbf{X}$ , nebo parametrem výkonostní funkce. Oba případy lze sjednotit zavedením funkce

$$\tilde{H}(\mathbf{U}; \boldsymbol{\theta}) = H(\mathbf{X}; \boldsymbol{\theta}_2) = H(F^{-1}(\mathbf{U}; \boldsymbol{\theta}_1); \boldsymbol{\theta}_2),$$

kde  $F^{-1}(\mathbf{U}; \boldsymbol{\theta}_1)$  je funkce inverzní ke kumulativní distribuční funkci  $\mathbf{X}$  a  $\mathbf{U}$  je náhodný vektor, jehož složky pocházejí z rovnoměrného rozdělení  $\mathcal{U}(0, 1)$ . Nyní lze psát

$$\frac{\partial \ell(\boldsymbol{\theta})}{\partial \theta_i} \approx \mathbb{E} \left( \frac{Y - Z}{\delta} \right),$$

kde  $Y = \tilde{H}(\mathbf{U}; \boldsymbol{\theta} + \frac{\delta}{2} \mathbf{e}_i)$  a  $Z = \tilde{H}(\mathbf{U}; \boldsymbol{\theta} - \frac{\delta}{2} \mathbf{e}_i)$ . Nahrazení střední hodnoty výběrovým průměrem vede na následující simulační algoritmus pro odhad gradientu funkce  $\ell(\boldsymbol{\theta})$ .

**Algoritmus 5.2** (Metoda centrálních diferencí s využitím CRN). Vstupem algoritmu je vhodně zvolená hodnota  $\delta > 0$ . Pro všechna  $i \in \{1, \dots, m\}$  jsou dále provedeny následující kroky:

1. Vygenerování náhodného výběru  $\mathbf{U}_1, \dots, \mathbf{U}_N$  z  $\mathcal{U}(0, 1)$ .
2. Výpočet hodnot  $Y_k = \tilde{H}(\mathbf{U}_k; \boldsymbol{\theta} + \frac{\delta}{2} \mathbf{e}_i)$  a  $Z_k = \tilde{H}(\mathbf{U}_k; \boldsymbol{\theta} - \frac{\delta}{2} \mathbf{e}_i)$  pro všechna  $k \in \{1, \dots, N\}$ .
3. Odhad hodnoty  $\frac{\partial \ell}{\partial \theta_i}(\boldsymbol{\theta})$  výběrovým průměrem z náhodné veličiny

$$\frac{Y - Z}{\delta},$$

tedy hodnotou

$$\widehat{\frac{\partial \ell}{\partial \theta_i}}(\boldsymbol{\theta}) \approx \frac{1}{N} \sum_{k=1}^N \frac{Y_k - Z_k}{\delta}.$$

4. Odhad  $(1 - \alpha)$  100% intervalu spolehlivosti pro parametr  $\frac{\partial \ell}{\partial \theta_i}(\boldsymbol{\theta})$ .

Hledaným  $(1 - \alpha)$  100% intervalem spolehlivosti pro parametr  $\frac{\partial \ell}{\partial \theta_i}(\boldsymbol{\theta})$  je interval

$$\left( \widehat{\frac{\partial \ell}{\partial \theta_i}}(\boldsymbol{\theta}) - z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{N}}, \widehat{\frac{\partial \ell}{\partial \theta_i}}(\boldsymbol{\theta}) + z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{N}} \right),$$

kde

$$\sigma = \frac{1}{\delta} \sqrt{\text{Var}(Y) + \text{Var}(Z) - 2 \text{Cov}(Y, Z)},$$

odvození je podobné jako v případě metody antitetických náhodných veličin. Odhad intervalu spolehlivosti lze opět získat nahrazením neznámých hodnot rozptylu a kovariance odpovídajícími výběrovými charakteristikami.

Pomocí algoritmu 5.2 bude dále nalezen odhad gradientu funkce  $\ell$  z příkladu 3.4.

**Příklad 5.3** (PERT, část 6). Projekt je složen ze sedmi činností, doba trvání  $i$ -té činnosti pochází z exponenciálního rozdělení s parametrem  $\theta_i = i^{-1}$ . Odhadovaná doba trvání projektu  $\ell$  může být chápána jako funkce vektoru  $\boldsymbol{\theta} = (\theta_1, \dots, \theta_7)$ , tedy  $\ell = \ell(\boldsymbol{\theta})$ .

Pro odhad gradientu funkce  $\ell(\boldsymbol{\theta})$  v bodě  $\boldsymbol{\theta} = (1^{-1}, \dots, 7^{-1})$  byl použit algoritmus 5.2 s volbou  $\delta = 0.01$ . Výsledkem simulace o rozsahu  $N = 10^6$  pokusů je odhad

$$\widehat{\nabla \ell}(\boldsymbol{\theta}) = (-0.185, -1.012, -7.26, -11.324, -10.288, -28.929, -21.883)^T,$$

mějme na paměti, že se nejedná o nestranný odhad. Ostatní výsledky, například jednotlivé  $(1 - \alpha) 100\%$  intervaly spolehlivosti, jsou uvedeny v tabulce B.1.

Získaný odhad gradientu je třeba správně interpretovat. Pro  $c \in \mathbb{R}$  blízké nule platí

$$\ell(\boldsymbol{\theta} + c\mathbf{e}_i) \approx \ell(\boldsymbol{\theta}) + c \left( \widehat{\nabla \ell}(\boldsymbol{\theta}) \right)_i.$$

Jednotlivé parciální derivace tedy vyjadřují vliv změny jednotlivých parametrů  $\theta_1, \dots, \theta_7$  o konstantu  $c$  na změnu odhadované doby trvání projektu. Z tohoto pohledu mají největší vliv na hodnotu  $\ell$  parametry  $\theta_5$  a  $\theta_6$ .

Pro identifikaci komponent s největším vlivem na hodnotu  $\ell$  je však vhodnější zkoumat vliv procentuální změny jednotlivých parametrů  $\theta_1, \dots, \theta_7$  na změnu  $\ell$ . Pro  $c \in \mathbb{R}$  blízké nule platí

$$\ell(\boldsymbol{\theta} + c\theta_i\mathbf{e}_i) \approx \ell(\boldsymbol{\theta}) + c\theta_i \cdot \left( \widehat{\nabla \ell}(\boldsymbol{\theta}) \right)_i,$$

což vede k myšlence identifikace důležitých komponent na základě vzájemného porovnání hodnot  $\theta_i \cdot \left( \widehat{\nabla \ell}(\boldsymbol{\theta}) \right)_i$  pro  $i \in \{1, \dots, 7\}$ . Pro jejich absolutní hodnoty platí

$$\left( \left| \theta_i \cdot \left( \widehat{\nabla \ell}(\boldsymbol{\theta}) \right)_i \right| \right)_{i=1, \dots, 7} = (0.185, 0.506, 2.420, 2.831, 2.058, 4.822, 3.126), \quad (5.1)$$

což ukazuje na vysoký vliv šesti komponenty. K podpoření této myšlenky nyní chápeme výslednou hodnotu  $\ell$  jako funkci vektoru  $\boldsymbol{\mu} = (\mu_1, \dots, \mu_7) = (\theta_1^{-1}, \dots, \theta_7^{-1})$ , tedy vektoru středních hodnot jednotlivých rozdělení pravděpodobnosti. Pro přehlednost bude použito označení  $\ell = \ell_2(\boldsymbol{\mu})$ , zřejmě platí  $\ell(\boldsymbol{\theta}) = \ell_2(\boldsymbol{\mu})$ . Jedná se pouze o změnu parametrizace hustoty pravděpodobnosti náhodného vektoru  $\mathbf{X}$ , což pochopitelně nemá vliv na důležitost jednotlivých komponent. Pomocí centrálních diferencí byl odhadnut gradient funkce  $\ell_2(\boldsymbol{\mu})$  v bodě  $\boldsymbol{\mu} = (1, \dots, 7)$ , výsledkem je vektor

$$\widehat{\nabla \ell_2}(\boldsymbol{\mu}) = (0.187, 0.253, 0.809, 0.704, 0.410, 0.804, 0.446)^T,$$

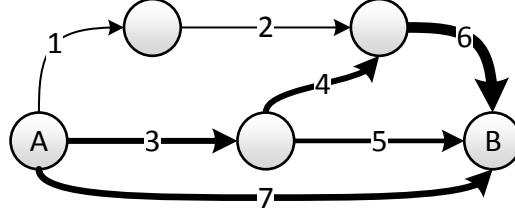
změna parametru  $\mu_i$  o konstantu má tedy největší vliv na hodnotu  $\ell$  v případě parametrů  $\mu_3, \mu_4$  a  $\mu_6$ . Další výsledky simulace jsou uvedeny v tabulce B.2. V tomto případě platí

$$\left( \left| \mu_i \cdot \left( \widehat{\nabla \ell_2}(\boldsymbol{\mu}) \right)_i \right| \right)_{i=1, \dots, 7} = (0.187, 0.506, 2.427, 2.817, 2.051, 4.825, 3.124), \quad (5.2)$$

což podle očekávání přibližně odpovídá vektoru (5.1).

Vzájemné porovnávání jednotlivých složek gradientu funkce  $\ell_2(\mu)$  by vedlo k identifikaci jiných komponent než v předchozím případě, zatímco porovnání hodnot vektoru (5.2) vede ke stejnému závěru jako porovnání hodnot vektoru (5.1).

Za komponentu s nejvyšším vlivem na hodnotu  $\ell$  lze považovat činnost 6. Vliv jednotlivých komponent ilustruje rovněž obrázek 5.1, širší spojnice značí větší důležitost komponenty.



Obrázek 5.1: Ilustrace vlivu jednotlivých komponent na hodnotu  $\ell$

△

## 5.2 Score Function

Metoda *score function*, dále SF, slouží k odhadu gradientu funkce  $\ell(\theta)$  v případě, že  $\theta$  je vektorem parametrů rozdělení náhodné veličiny  $\mathbf{X}$ , tedy

$$\ell(\theta) = \mathbb{E}_{\theta}(H(\mathbf{X})).$$

Pro gradient funkce  $\ell(\theta)$  platí

$$\nabla \ell(\theta) = \nabla_{\theta} \int H(\mathbf{x}) f(\mathbf{x}; \theta) \mu(d\mathbf{x}). \quad (5.3)$$

Tato metoda využívá větu 5.4, která za jistých podmínek umožňuje výměnu derivace a integrace v (5.3).

**Věta 5.4** (Výměna derivace a integrace, viz [2]). *Nechť*

- $g(\mathbf{x}; \theta)$  je diferencovatelná v bodě  $\theta_0 \in \mathbb{R}^k$ ,
- existuje  $\int \nabla_{\theta} g(\mathbf{x}; \theta_0) \mu(d\mathbf{x})$ ,
- existuje okolí  $U$  bodu  $\theta_0$  a integrovatelná funkce  $M(\mathbf{x}; \theta_0)$  taková, že pro všechna  $\theta \in U$

$$\frac{|g(\mathbf{x}; \theta) - g(\mathbf{x}; \theta_0)|}{\|\theta - \theta_0\|} \leq M(\mathbf{x}; \theta_0).$$

Pak

$$\nabla_{\theta} \int g(\mathbf{x}; \theta) \mu(d\mathbf{x}) \Big|_{\theta=\theta_0} = \int \nabla_{\theta} g(\mathbf{x}; \theta_0) \mu(d\mathbf{x}).$$

*Poznámka k důkazu.* Je-li  $\mu$  aritmetická míra, stačí integraci nahradit sumou. Ve spojitém případě důkaz vychází z Lebesgueovy věty, viz. [2, str. 422].

Jsou-li splněny předpoklady věty 5.4, platí

$$\begin{aligned}\nabla \ell(\boldsymbol{\theta}) &= \int H(\mathbf{x}) \nabla_{\boldsymbol{\theta}} f(\mathbf{x}; \boldsymbol{\theta}) \mu(d\mathbf{x}) = \\ &= \int H(\mathbf{x}) \frac{\nabla_{\boldsymbol{\theta}} f(\mathbf{x}; \boldsymbol{\theta})}{f(\mathbf{x}; \boldsymbol{\theta})} f(\mathbf{x}; \boldsymbol{\theta}) \mu(d\mathbf{x}) = \\ &= \mathbb{E}_{\boldsymbol{\theta}}(H(\mathbf{X}) \mathcal{S}(\boldsymbol{\theta}; \mathbf{X})),\end{aligned}$$

kde

$$\mathcal{S}(\boldsymbol{\theta}; \mathbf{x}) = \frac{\nabla_{\boldsymbol{\theta}} f(\mathbf{x}; \boldsymbol{\theta})}{f(\mathbf{x}; \boldsymbol{\theta})} = \nabla_{\boldsymbol{\theta}} \ln f(\mathbf{x}; \boldsymbol{\theta})$$

je tzv. *score function*. Nahrazení střední hodnoty výběrovým průměrem vede na algoritmus 5.5.

---

**Algoritmus 5.5** (Odhad gradientu pomocí metody SF).

1. Vygenerování náhodného výběru  $\mathbf{X}_1, \dots, \mathbf{X}_N$  z  $f(\cdot; \boldsymbol{\theta})$ .
2. Odhad  $\nabla \ell(\boldsymbol{\theta})$  výběrovým průměrem z náhodné veličiny  $H(\mathbf{X}) \mathcal{S}(\boldsymbol{\theta}; \mathbf{X})$ , tedy hodnotou

$$\widehat{\nabla} \ell(\boldsymbol{\theta}) = \frac{1}{N} \sum_{k=1}^N H(\mathbf{X}_k) \mathcal{S}(\boldsymbol{\theta}; \mathbf{X}_k). \quad (5.4)$$

3. Odhad  $(1 - \alpha)$  100% intervalů spolehlivosti pro jednotlivé složky vektoru  $\nabla \ell(\boldsymbol{\theta})$ .
- 

Výhodou oproti metodě centrálních diferencí (viz algoritmus 5.2) je nestrannost estimatoru (5.4).

### 5.2.1 Kombinace metod SF a IS

V předchozí kapitole byla metoda IS představena jako metoda redukce rozptylu, její využití je však širší, viz [2]. Základní metodu IS, viz algoritmus 4.17, lze použít pro odhad hodnot funkce  $\ell(\boldsymbol{\theta})$  ve více různých bodech  $\boldsymbol{\theta} \in \Theta$  během jedné simulace. S použitím principu metody IS lze zapsat funkci  $\ell(\boldsymbol{\theta})$  jako

$$\ell(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\eta}} H(\mathbf{X}) \frac{f(\mathbf{X}; \boldsymbol{\theta})}{f(\mathbf{X}; \boldsymbol{\eta})},$$

kde  $\boldsymbol{\eta} \in \Theta$  a hustota  $f(\mathbf{X}; \boldsymbol{\eta})$  splňuje podmínku (4.8). Střední hodnotu lze nahradit výběrovým průměrem a odhadovat hodnotu  $\ell(\boldsymbol{\theta})$  simulačně na základě náhodného výběru  $\mathbf{X}_1, \dots, \mathbf{X}_N$  z rozdělení pravděpodobnosti o hustotě  $f(\mathbf{X}; \boldsymbol{\eta})$ . Stejný náhodný výběr je

možné použít pro odhad hodnoty funkce  $\ell(\theta)$  ve více bodech  $\theta \in \Theta$ . Je-li určována hodnota  $\ell(\theta)$  v  $M$  různých bodech  $\theta \in \{\theta_1, \dots, \theta_M\} \subset \Theta$ , je výsledkem simulace vektor  $(\widehat{\ell}(\theta_1), \dots, \widehat{\ell}(\theta_M))$ .

Podobně je možné gradient funkce  $\ell(\theta)$  zapsat ve tvaru

$$\begin{aligned}\nabla \ell(\theta) &= \int H(x) \frac{\nabla_{\theta} f(x; \theta)}{f(x; \theta)} \frac{f(x; \theta)}{f(x; \eta)} f(x; \eta) \mu(dx) = \\ &= \mathbb{E}_{\eta}(H(X) S(\theta; X) W(X; \theta, \eta)),\end{aligned}$$

kde  $\eta \in \Theta$ ,  $W(X; \theta, \eta) = f(x; \theta) / f(x; \eta)$  a hustota  $f(X; \eta)$  splňuje podmínku (4.8). Tento zápis vede na následující algoritmus pro odhad gradientu funkce  $\ell(\theta)$  v bodě  $\theta$ , viz [1].

---

**Algoritmus 5.6** (Odhad gradientu pomocí metody SF s využitím IS).

1. Vygenerování náhodného výběru  $X_1, \dots, X_N$  z  $f(\cdot; \eta)$ .
2. Odhad  $\nabla_{\theta} \ell(\theta)$  výběrovým průměrem z náhodné veličiny

$$H(X) S(\theta; X) W(X; \theta, \eta),$$

tedy hodnotou

$$\widehat{\nabla} \ell(\theta) = \frac{1}{N} \sum_{k=1}^N H(X_k) S(\theta; X_k) W(X_k; \theta, \eta).$$

3. Odhad  $(1 - \alpha)$  100% intervalů spolehlivosti pro jednotlivé složky vektoru  $\nabla \ell(\theta)$ .
- 

Algoritmus 5.6 lze rovněž využít pro odhad gradientu funkce  $\ell(\theta)$  ve více různých bodech  $\theta \in \Theta$  během jedné simulace. V tomto případě je beze změny proveden první krok algoritmu 5.6, přičemž IS hustota může být volena s ohledem na redukci rozptylu, rovněž však lze ponechat původní hustotu pravděpodobnosti náhodné veličiny  $X$ . Kroky 2. a 3. jsou provedeny zároveň pro  $M$  různých bodů  $\theta \in \{\theta_1, \dots, \theta_M\}$  s použitím téhož náhodného výběru  $X_1, \dots, X_N$  z  $f(\cdot; \eta)$ . Výsledkem simulace je matice hodnot

$$(\widehat{\nabla} \ell(\theta_1), \dots, \widehat{\nabla} \ell(\theta_M)).$$

Odhad gradientu funkce  $\ell$  z příkladu 3.4 bude nyní nalezen pomocí metody SF, nejprve pomocí základního algoritmu, poté s využitím metody IS.

**Příklad 5.7** (PERT, část 7). Prvním úkolem je odhadnout gradient  $\nabla \ell(\theta)$  pomocí metody SF, je tedy potřeba určit funkci  $S(\theta; x) = \nabla_{\theta} \ln f(x; \theta)$ , kde

$$f(x; \theta) = \prod_{k=1}^7 (\theta_k e^{-\theta_k x_k}).$$

Pro parciální derivace funkce  $\ln f(\mathbf{x}; \boldsymbol{\theta})$  podle jednotlivých složek vektoru  $\boldsymbol{\theta} = (\theta_1, \dots, \theta_7)$  platí

$$\frac{\partial}{\partial \theta_i} \ln f(\mathbf{x}; \boldsymbol{\theta}) = \frac{\partial}{\partial \theta_i} \sum_{k=1}^7 (\ln \theta_k - \theta_k x_k) = \frac{1}{\theta_i} - x_i,$$

proto

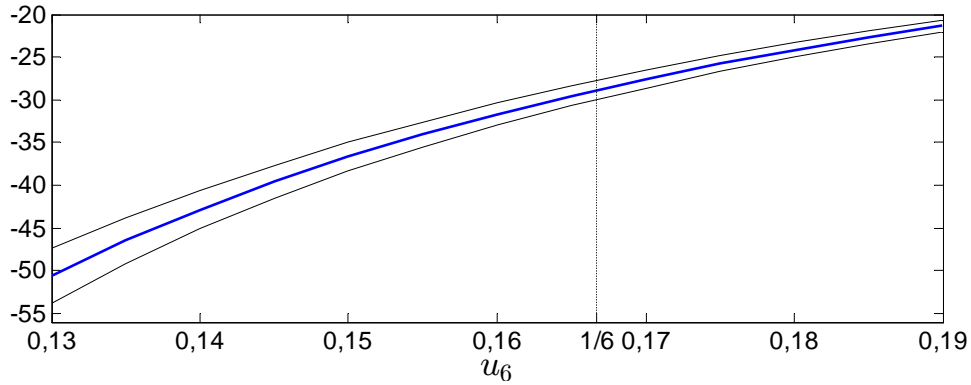
$$\mathcal{S}(\boldsymbol{\theta}; \mathbf{x}) = \left( \frac{1}{\theta_1} - x_1, \dots, \frac{1}{\theta_7} - x_7 \right).$$

Pro odhad  $\nabla \ell(\boldsymbol{\theta})$  v bodě  $\boldsymbol{\theta} = (1^{-1}, \dots, 7^{-1})$  byl použit algoritmus 5.5, bylo provedeno  $N = 10^6$  náhodných pokusů. Výsledkem je

$$\widehat{\nabla \ell}(\boldsymbol{\theta}) = (-0.197, -1.001, -7.289, -11.404, -10.195, -28.809, -22.417)^T,$$

což přibližně odpovídá řešení získanému pomocí metody centrálních diferencí. Přesnost odhadu je v tomto případě nižší, výhodou je však jeho nestrannost, jednotlivé  $(1 - \alpha)$  100% intervaly spolehlivosti jsou uvedeny v tabulce B.3.

S využitím metody IS byl pro ilustraci odhadnut průběh funkce  $\frac{\partial \ell}{\partial \theta_6}(\boldsymbol{\theta})$ . Byl použit algoritmus 5.6 s volbou  $\boldsymbol{\theta}_0 = (1^{-1}, \dots, 7^{-1})$ , hodnota  $\frac{\partial \ell}{\partial \theta_6}(\boldsymbol{\theta})$  byla odhadována v bodech  $\boldsymbol{\theta} = (1^{-1}, \dots, 5^{-1}, \theta_6, 7^{-1})$ , kde  $\theta_6 \in \{0.130, 0.135, \dots, 0.190\}$ . Obrázek 5.2 znázorňuje získaný odhad průběhu funkce  $\frac{\partial \ell}{\partial \theta_6}(\boldsymbol{\theta})$ , je chápána jako funkce proměnné  $\theta_6$ , neboť ostatní složky vektoru  $\boldsymbol{\theta}$  jsou fixní. V grafu je dále zvýrazněn bod  $\theta_6 = 6^{-1}$ , který je parametrem rozdělení náhodné veličiny  $X_6$ , a 95% meze spolehlivosti.

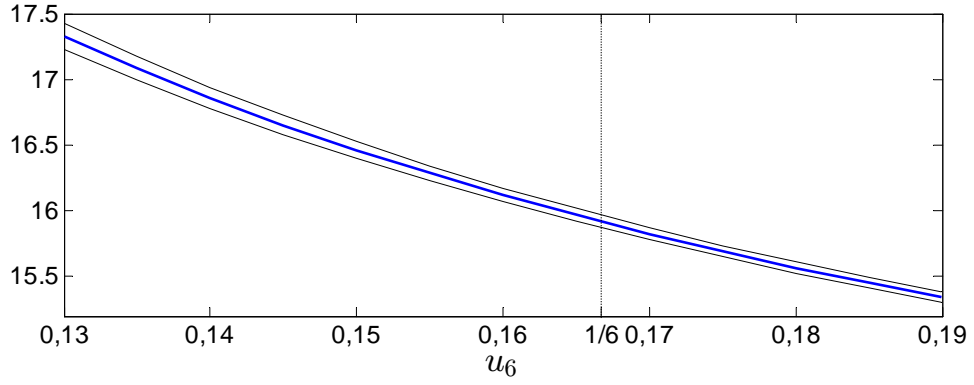


Obrázek 5.2: Odhad průběhu funkce  $(\nabla \ell(\boldsymbol{\theta}))_6$  pomocí metody SF

Pomocí metody IS byly odhadnuty rovněž hodnoty funkce  $\ell(\boldsymbol{\theta})$  ve stejných bodech. Získaný odhad včetně 95% mezí spolehlivosti znázorňuje graf na obrázku 5.3.

Pro porovnání s příkladem 5.3 bude dále odhadnut gradient funkce  $\ell_2(\boldsymbol{\mu})$  v bodě  $\boldsymbol{\mu} = (1, \dots, 7)$ , je tedy potřeba určit funkci  $\mathcal{S}_2(\boldsymbol{\mu}; \mathbf{x}) = \nabla_{\boldsymbol{\mu}} \ln f_2(\mathbf{x}; \boldsymbol{\mu})$ , kde

$$f_2(\mathbf{x}; \boldsymbol{\mu}) = \prod_{k=1}^7 \left( \frac{1}{\mu_k} e^{-\frac{x_k}{\mu_k}} \right).$$



Obrázek 5.3: Odhad průběhu funkce  $\ell(\theta)$  pomocí metody IS

V tomto případě platí

$$\frac{\partial}{\partial \mu_i} \ln f_2(\mathbf{x}; \boldsymbol{\mu}) = \frac{\partial}{\partial \mu_i} \sum_{k=1}^7 \left( \ln \frac{1}{\mu_k} - \frac{x_k}{\mu_k} \right) = -\frac{1}{\mu_i} + \frac{x_i}{\mu_i^2},$$

tedy

$$\mathcal{S}_2(\boldsymbol{\mu}; \mathbf{x}) = \left( -\frac{1}{\mu_1} + \frac{x_1}{\mu_1^2}, \dots, -\frac{1}{\mu_7} + \frac{x_7}{\mu_7^2} \right).$$

Odhadem gradientu funkce  $\ell_2(\boldsymbol{\mu})$  v bodě  $\boldsymbol{\mu} = (1, \dots, 7)$  pomocí metody SF je

$$\widehat{\nabla \ell_2}(\boldsymbol{\mu}) = (0.174, 0.252, 0.808, 0.713, 0.414, 0.809, 0.447)^T,$$

což přibližně odpovídá hodnotám získaným pomocí metody centrálních diferencí. Přesnost tohoto odhadu je uvedena v tabulce B.4.  $\triangle$

## 6 Implementace a paralelizace

Veškeré metody redukce rozptylu a analýzy citlivosti popsané v předchozích kapitolách byly implementovány v jazyce Matlab, konkrétně se jedná o metody

- základní a podmíněná metoda Monte Carlo, viz algoritmy 3.3 a 4.14,
- metoda antitetických náhodných veličin, viz algoritmus 4.4,
- metoda řídících veličin, viz algoritmus 4.8,
- metoda IS včetně metod pro odhad parametrů IS hustoty, viz 4.17, 4.23 a 4.26
- metoda centrálních diferencí, viz. algoritmus 5.2,
- metoda SF a její kombinace s metodou IS, viz. algoritmy 5.5 a 5.6.

Pro tyto algoritmy byla implementována obecná schémata, která lze bez modifikací použít v mnoha aplikacích.

Pro ilustraci nyní popíšeme nejjednodušší z těchto schémat, a to kód metody MC, který se shoduje s kódem podmíněné metody MC, viz 6.1. Tyto dvě metody se liší pouze použitou výkonnostní funkcí.

---

```

1  function [ l,ss ] = Estimator_MC(theta,N,H,rnd,batch_size)
2  %ESTIMATOR_MC - algoritmus zakladni metody MC
3  %   theta      vektor parametru rozdeleni
4  %   N          pocet nahodnych pokusu
5  %   H          vykonnostni funkce
6  %   rnd        funkce generujici nahodna cisla
7  %   batch_size velikost davky (volitelny parametr)
8  if nargin==4
9      batch_size=1e6;
10 end
11 N2=batch_size*ones(1,ceil(N/batch_size)); % pole velikosti davek
12 if mod(N,batch_size)~=0
13     N2(end)=mod(N,batch_size);
14 end
15 sumY=0; % suma vysledku vykonnostni funkce
16 sumYY=0; % suma druhych mocnin vysledku
17 for i=1:length(N2)
18     Ni=N2(i);
19     X=rnd(repmat(theta,Ni,1)); % vygenerovani davky nahodnych cisel
20     Y=H(X); % aplikace vykonnostni funkce na davku
21     sumY=sumY+sum(Y);
22     sumYY=sumYY+sum(Y.*Y);
23 end
24 l=sumY/N; % vyberovy prumer
25 ss=(sumYY-sumY*sumY/N)/(N-1); % vyberovy rozptyl

```

---

Výpis 6.1: Obecné schéma metody MC



Je-li odhadována hodnota  $\ell = \mathbb{E}(H(\mathbf{X}))$ , kde  $\mathbf{X}$  je pochází z  $n$ -rozměrného rozdělení pravděpodobnosti o hustotě  $f$ , je prvním vstupem vektor parametrů tohoto rozdělení o délce  $n$ . Druhým vstupem je požadovaný rozsah náhodného výběru. Třetím vstupem je funkce, která pro matici parametrů o rozměrech  $N \times n$  vrací matici náhodných čísel vygenerovanou z rozdělení pravděpodobnosti o hustotě  $f$  s příslušnými parametry. Čtvrtým vstupem je výkonnostní funkce, která pro matici náhodných hodnot o rozměrech  $N \times n$  vrací sloupcový vektor výsledků výkonnostní funkce o délce  $N$ .

Generování náhodných čísel a následné vyhodnocení systémové funkce je kvůli snížení paměťových nároků rozděleno do několika dávek. Podobné kódy pro ostatní metody lze nalézt v příloze na CD, metody pracují i s dalšími vstupy, v případě metody IS je vstupem například hustota pravděpodobnosti a vektor indexů parametrů, které mají být zahrnuty do LR.

Výstupem metod redukce rozptylu a metod pro odhad gradientu je vždy odhad hledané hodnoty vypočtený jako výběrový průměr a rozptyl tohoto odhadu, který lze použít pro určení intervalu spolehlivosti a dalších měřítek přesnosti odhadu.

Samotná schémata metod nejsou výpočetně náročná, čas simulace je dán implementací vstupních funkcí, zejména výkonnostní funkce  $H(\mathbf{X})$ . Tyto funkce lze efektivně implementovat například s využitím technologie CUDA (z anglického *Compute Unified Device Architecture*), která umožňuje vykonávání paralelních výpočtů na grafických kartách značky NVIDIA. Podrobnosti o technologii CUDA lze nalézt v [10].

## 6.1 Paralelizace výkonnostní funkce pomocí CUDA

Princip paralelizace výkonnostní funkce je snadný, neboť v případě metody MC a příbuzných metod je tato funkce vyhodnocována v  $N$  nezávislých pokusech. Požadovaných  $N$  pokusů stačí rozdělit mezi jednotlivá nezávislá vlákna, mezi nimiž neprobíhá žádná komunikace. Tímto způsobem lze tedy nahradit libovolný cyklus s pevným počtem opakování, jehož jednotlivá opakování jsou nezávislá a nezáleží tak na pořadí jejich vykonání.

Jednotlivá vlákna jsou organizována do bloků a bloky dále tvoří mřížku, maximální počet vláken v bloku je obvykle 1024. Kód algoritmu, který má být proveden v každém pokusu, je zapsán v tzv. CUDA kernelu, syntaxe je obdobou jazyka C. Tato funkce je spuštěna ve všech vláknech, jednotlivá vlákna je možné rozlišit pomocí indexu vlákna v bloku a indexu bloku. Počet vláken, která jsou zpracovávána zároveň, závisí na počtu jader dané GPU.

Použití technologie CUDA usnadňuje fakt, že s CUDA kernely je možné pracovat přímo z prostředí Matlab, viz [4]. Nejprve je třeba vytvořit objekt kernel a nastavit jeho vlastnosti, zejména specifikovat rozměry mřížky a bloků a určit tak počet vláken, v nichž bude CUDA kernel vyhodnocován. Dalším krokem je uložení dat, se kterými funkce kernel pracuje, na GPU. Je možné pracovat s více typy paměti, obvykle jsou data ukládána do globální paměti, kam mají přístup všechna vlákna. Pro data, která se v průběhu výpočtu nemění, lze použít rovněž konstantní paměť. Následuje spuštění samotného kernelu. Jednotlivá vlákna používají kromě již zmíněných typů paměti zejména paměť lokální. Mimoto existuje také sdílená paměť určená ke komunikaci mezi jednotlivými vlákny

v bloku, v případě MC simulací ji však nebylo třeba používat. Po vykonání výpočtu na GPU zbývá přesunout výsledky výpočtu z globální paměti GPU do operační paměti a přejít k jejich vyhodnocení.

Příkladem kódu paralelizované výkonnostní funkce je 6.2, tento kód přímo následuje výše popsané kroky. Jedná se o výkonnostní funkci, která bude použita v kapitole 8 při odhadu spolehlivosti systémů s nezávislými prvky.

---

```

1  function [ Y ] = H( X,M )
2  %H - vykonnostni funkce pouzivana v kapitole 8
3  %X...vstupni matice nahodnych cisel o rozmerech N x n
4  N=size(X,1);           % pocet pozadovanych nahodnych pokusu
5  n=size(X,2);           % pocet komponent
6  %% vytvoreni objektu kernel a nastaveni jeho vlastnosti
7  k=parallel.gpu.CUDAKernel('H_CUDA.ptx','H_CUDA.cu');
8  b_size=min(N,1e3);      % pocet vlaken na blok
9  k.ThreadBlockSize=[b_size 1 1]; % nastaveni rozmeru rozmeru bloku
10 g_size=ceil(min(N/b_size,5e3)); % pocet bloku
11 k.GridSize=[g_size 1];  % nastaveni rozmeru mridky
12 N0=b_size*g_size;       % pocet provedenych pokusu (N0>=N)
13 %% ulozeni dat, se kterymi kernel pracuje, na GPU
14 X=X'; X=gpuArray(single(X(:))); % ulozeni stavovych vektoru na GPU
15 results=gpuArray(zeros(1,N0,'int32')); % alokace vystupniho vektoru
16 A=prevod_mat(M,32);     % vstupni data ulohy
17 %% spusteni kernelu a stazeni vysledku
18 [~,results,~]=feval(k,n,X,results,A); % spusteni kernelu
19 Y=double((gather(results(1:N)))'); % vystupni vektor
20 end

```

---

Výpis 6.2: Paralelizovaná výkonnostní funkce

Výhodou je snadná přenositelnost kódu, vytvořený CUDA kernel může být v nezměněné podobě spuštěn na jiné grafické kartě. Je vhodné pouze přizpůsobit rozměry mřížky a bloků počtu multiprocesorů, což nevyžaduje změnu CUDA kernelu. Při testování uvedeném v kapitolách 7 a 8 byla použita grafická karta NVIDIA GT 555M, je určena pro notebooky a má 144 jader. Výsledky ukazují, že na běžném počítači s grafickou kartou podporující technologii CUDA lze dosáhnout značného snížení výpočetního času oproti ekvivalentní implementaci v jazyce Matlab určené pro CPU. Pro srovnání efektivita CPU výpočtů byla testována na procesoru Intel Core i7 - 2630QM.

Paralelizací konkrétních algoritmů se zabývají sekce 7.3 a 8.4 následujících kapitol. Je zde rovněž porovnávána efektivita CUDA implementace se sekvenční implementací a s paralelní implementací využívající cyklus `parfor` v jazyce Matlab. V příloze B jsou uvedeny příklady kódů funkcí kernel a výsledky testování.

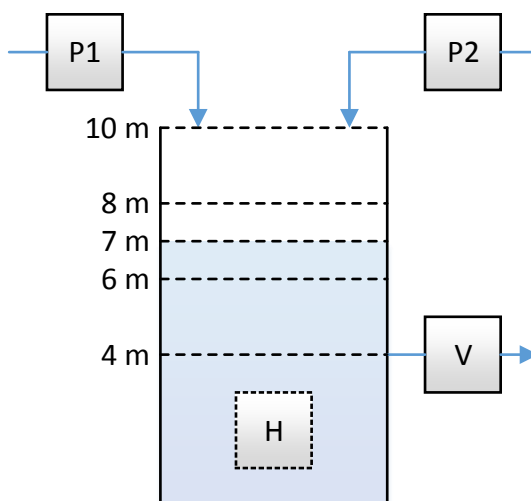
## 7 Aplikace I - dynamický systém se dvěma procesními proměnnými

Poznatky z předchozích kapitol budou nyní aplikovány na řešení problému dynamické spolehlivosti, pocházejícího z [11]. Tento testovací dynamický systém se v literatuře vyskytuje v mnoha variantách, liší se například uvažováním jedné nebo dvou procesních proměnných a závislostí či nezávislostí stochastických vlastností systému na procesních proměnných. Pro aplikaci metod popsaných v předchozích kapitolách byla vybrána varianta se dvěma procesními proměnnými a se závislostí stochastických vlastností na jedné z procesních proměnných.

V dostupné literatuře byla simulace tohoto dynamického systému, či některé z jeho variant, provedena různými metodami, například pomocí Petriho sítí [12, 13], nebo pomocí Bayesovských sítí [14]. V této kapitole bude tentýž systém simulován metodami založenými na MC, to znamená, že bude vždy vykonáno více běhů simulace daného systému a ty budou následně vyhodnoceny. Cíle jsou tedy dva, prvním je navrhnout efektivní algoritmus pro vykonání běhu simulace a druhým pomocí metod redukce rozptylu zajistit, aby těchto běhů simulace stačilo pro dosažení určité přesnosti vykonat co nejméně.

### 7.1 Specifikace systému

Systém je tvořen nádrží s tekutinou, například vodou, dvěma čerpadly, ventilem a ohřívačem. Čerpadla zajišťují přítok vody do nádrže a ventil odtok z nádrže. První procesní proměnnou je teplota vody v nádrži, její hodnotu v čase  $t \geq 0$  udává funkce  $\Theta(t)$ , přitékající voda má teplotu  $\Theta_{in}$ . Druhou procesní proměnnou je výška hladiny, jejíž hodnotu udává funkce  $H(t)$ . Dále bude výška hladiny uváděna v metrech, teplota vody ve stupních Celsia a čas v hodinách. Schéma tohoto systému je znázorněno na obrázku 7.1.



Obrázek 7.1: Schéma systému

Model systému se skládá ze tří komponent, konkrétně dvou čerpadel (značeny P1 a P2 z anglického *pump*) a jednoho ventilu (značen V z anglického *valve*). Všechny komponenty mají stejnou průtokovou rychlost  $Q$ . Předpokládejme, že vodorovný průřez nádrže je konstantní, průtokovou rychlost komponent je tedy možné udávat ve výškových metrech za hodinu. Součástí systému je rovněž ohřívač vody (značen H z anglického *heater*), není však nutné jej považovat za komponentu, neboť v čase nemění svůj stav (je stále v provozu). Pro jednoduchost dále předpokládáme, že je voda ohřívána rovnoměrně v celém objemu. Rychlost ohřevu vody  $w$  má jednotku  $[\text{m} \cdot ^\circ\text{C} \cdot \text{h}^{-1}]$ , tedy pokud  $w = 1$ , ohřeje se 1 výškový metr vody o 1 stupeň za 1 hodinu.

Každá z komponent se může v daném čase  $t \geq 0$  nacházet v jenom ze čtyř stavů,

- otevřená a zároveň v provozu,
- zavřená a zároveň v provozu,
- otevřená a zároveň mimo provoz,
- zavřená a zároveň mimo provoz.

Z hlediska klasifikace systémů se jedná o DE systém, neboť k poruchám jednotlivých komponent dochází v diskrétních časových okamžicích. Procesní proměnné teplota a výška hladiny se mění spojitě, jejich vývoj je však deterministický.

Přiřaďme nyní komponentám P1, P2 a V postupně indikátory  $a_1$ ,  $a_2$  a  $a_3$ . Platí  $a_i = 1$ , je-li odpovídající komponenta otevřená a  $a_i = 0$ , je-li zavřená. Se zavedeným značením lze vztah mezi funkcemi  $H(t)$  a  $\Theta(t)$  popsat pomocí diferenciálních rovnic

$$\frac{d}{dt}H(t) = (a_1 + a_2 - a_3)Q, \quad (7.1)$$

$$\frac{d}{dt}[H(t)\Theta(t)] = a_1Q\Theta_{in} + a_2Q\Theta_{in} - a_3Q\Theta(t) + w. \quad (7.2)$$

Z rovnice (7.1) plyne

$$H(t) = H(0) + (a_1 + a_2 - a_3)Qt. \quad (7.3)$$

Řešení rovnice (7.2) se liší v závislosti na hodnotách indikátorů  $a_i$ . Výsledné funkce  $\Theta(t)$  pro jednotlivé konfigurace systému jsou uvedeny v tabulce 7.1.

Náhodná veličina doba do poruchy  $i$ -té komponenty pochází z rozdělení pravděpodobnosti s distribuční funkcí

$$F_i(t) = 1 - \exp\left(-\lambda_i \int_0^t b(\Theta(\tau)) d\tau\right), \quad (7.4)$$

kde

$$b(x) = \frac{4}{5} \exp[0.05756 \cdot (x - 20)] + \frac{1}{5} \exp[-0.2301 \cdot (x - 20)].$$

Dojde-li k poruše komponenty, s pravděpodobností 0.5 bude po porouchání otevřená ( $a_i = 1$ ) a s pravděpodobností 0.5 zavřená ( $a_i = 0$ ). Opravy komponent nejsou uvažovány, stav porouchané komponenty již tedy nemůže být změněn.

$a_1$	$a_2$	$a_3$	$\Theta(t)$
0	0	0	$\Theta(t) = \Theta(0) + \frac{w}{H(0)}t$
0	0	1	$\Theta(t) = \Theta(0) - \frac{w}{Q} \ln \left[ 1 - \frac{Q}{H(0)}t \right]$
0	1	0	$\Theta(t) = \Theta_m + \frac{w}{Q} + \left( \Theta(0) - \Theta_m - \frac{w}{Q} \right) \cdot \left[ 1 + \frac{Q}{H(0)}t \right]^{-1}$
1	0	0	
0	1	1	$\Theta(t) = \left[ \Theta_m + \frac{w}{Q} \right] + \left( \Theta(0) - \Theta_m - \frac{w}{Q} \right) \cdot \exp \left( -\frac{Q}{H(0)}t \right)$
1	0	1	
1	1	0	$\Theta(t) = \left[ \Theta_m + \frac{w}{2Q} \right] + \left( \Theta(0) - \Theta_m - \frac{w}{2Q} \right) \cdot \left[ 1 + \frac{2Q}{H(0)}t \right]^{-1}$
1	1	1	$\Theta(t) = \left[ \Theta_m + \frac{w}{2Q} \right] + \left( \Theta(0) - \Theta_m - \frac{w}{2Q} \right) \cdot \left[ 1 + \frac{Q}{H(0)}t \right]^{-2}$

Tabulka 7.1: Funkce  $\Theta(t)$  v závislosti na konfiguraci systému

V důsledku poruch jednotlivých komponent mohou nastat tři různé poruchy systému, přetečení, vysušení a vysoká teplota. Počáteční hodnota procesní proměnné výška hladiny je 7 m. K přetečení dojde, pokud výška hladiny dosáhne 10 m, k vysušení dojde, pokud dosáhne 4 m. Porucha vysoká teplota nastane v případě, že teplota vody v nádrži dosáhne 100 °C. V čase  $t = 0$  jsou všechny komponenty v provozu, přičemž komponenty P1 a V jsou otevřené a P2 zavřená, dokud tedy nedojde k poruše některé z komponent, nezmění se ani množství vody v nádrži.

Vedle počáteční výšky hladiny a výšek hladiny, kdy dojde k přetečení, jsou ve schématu 7.1 vyznačeny dvě další významné hodnoty hladiny. Výška 6 metrů značí riziko vysušení a výška 8 metrů riziko přetečení. Systém se řídí dvěma jednoduchými pravidly,

- pokud výška hladiny dosáhne 6 metrů, obě čerpadla se otevřou a ventil zavře (je-li to možné),
- pokud výška hladiny dosáhne 8 metrů, obě čerpadla se zavřou a ventil otevře (je-li to možné).

## 7.2 Simulace systému

Úloha byla řešena pro parametry uvedené v tabulce 7.2. Jako počáteční teplota byla záměrně zvolena hodnota, na níž se teplota vody ustálí, je-li otevřeno právě jedno čerpadlo.

$\Theta(0)$	$H(0)$	$Q$	$\Theta_{in}$	$w$	$\lambda_1$	$\lambda_2$	$\lambda_3$
15.6	7	1.5	15	1	$\frac{1}{219}$	$\frac{1}{175}$	$\frac{1}{320}$

Tabulka 7.2: Parametry úlohy

Systém byl simulován v konečném časovém horizontu, konkrétně v časovém intervalu  $\langle 0, T \rangle$ , kde  $T = 1000$ . Způsob simulace vychází z mechanismu *event clock*, viz algoritmus 2.1. V průběhu simulace může nastat několik typů událostí. První tři typy jsou zároveň poruchami systému, při jejich nastání simulace končí, jedná se o události

- **vysušení**, jestliže výška hladiny dosáhne 4 m,
- **přetečení**, jestliže výška hladiny dosáhne 10 m,
- **vysoká teplota**, jestliže teplota vody dosáhne 100 °C.

Dále mohou nastat události

- **riziko vysušení**, jestliže výška hladiny dosáhne 6 m,
- **riziko přetečení**, jestliže výška hladiny dosáhne 8 m,

při nastání některé z těchto událostí je konfigurace systému upravena dle řídicích pravidel. Posledním typem události je porucha některé z dosud neporouchaných komponent, jedná se o události

- **porucha komponenty P1**,
- **porucha komponenty P2**,
- **porucha komponenty V**.

Od nastání poruchy některé z komponent bude tato komponenta s pravděpodobností 0.5 až do konce simulace otevřená a s pravděpodobností 0.5 zavřená.

Oproti běžnému mechanismu *event clock* je zde při nastání určité události nutné celý seznam událostí smazat a události naplánovat znovu. Simulace se tedy řídí následujícím algoritmem.

---

**Algoritmus 7.1** (Event clock pro řešení problému dynamické spolehlivosti).

1. Vytvořit počáteční seznam událostí naplánováním poruch komponent P1, P2, V.
  2. Seřadit seznam událostí podle času událostí.
  3. Provést první událost na seznamu, nastavit proměnnou  $t$  na čas této události.
  4. Vytvořit nový seznam událostí naplánováním události způsobené výškou hladiny, události vysoká teplota a událostí způsobených poruchami dosud neporouchaných komponent.
  5. Není-li seznam událostí prázdný, pokračovat krokem 2.
-

Při plánování události způsobené výškou hladiny je nejprve vypočtena aktuální rychlost přítoku

$$Q_a = (a_1 + a_2 - a_3) Q.$$

V závislost na tom, zda  $Q_a > 0$ , nebo  $Q_a < 0$ , je vyhodnoceno, která z událostí souvisejících s výškou hladiny (vysušení, riziko vysušení, přetečení, riziko přetečení) bude následovat, a je vypočten její čas pomocí inverzní funkce k funkci  $H(t)$ . Pokud  $Q_a = 0$ , není naplánována žádná událost související s výškou hladiny.

Podobně při plánování události vysoká teplota je z tabulky 7.1 vybrána funkce  $\Theta(t)$  odpovídající aktuální konfiguraci a pomocí její inverze je určen čas této události.

Naplánování poruchy  $i$ -té komponenty spočívá ve vygenerování času poruchy z rozdělení pravděpodobnosti s distribuční funkcí (7.4). K tomu byla použita inverzní transformace, byla tedy vygenerována hodnota  $\xi_i \sim \mathcal{U}(0, 1)$  a k ní byla dále hledána odpovídající hodnota  $t_i$  taková, že

$$\xi_i \approx 1 - \exp \left( -\lambda_i \int_0^{t_i} b(\Theta(\tau)) d\tau \right).$$

Určitý integrál, jehož horní mez  $t_i$  je úkolem odhadnout, byl řešen numericky s použitím obdélníkového pravidla. Jelikož zde je neznámou integrační mez, vede tento přístup na následující algoritmus.

---

**Algoritmus 7.2** (Odhad horní meze určitého integrálu pomocí obdélníkového pravidla).

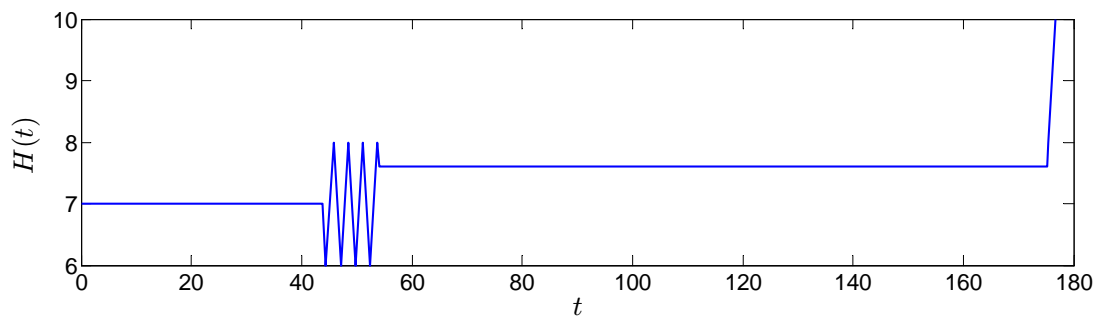
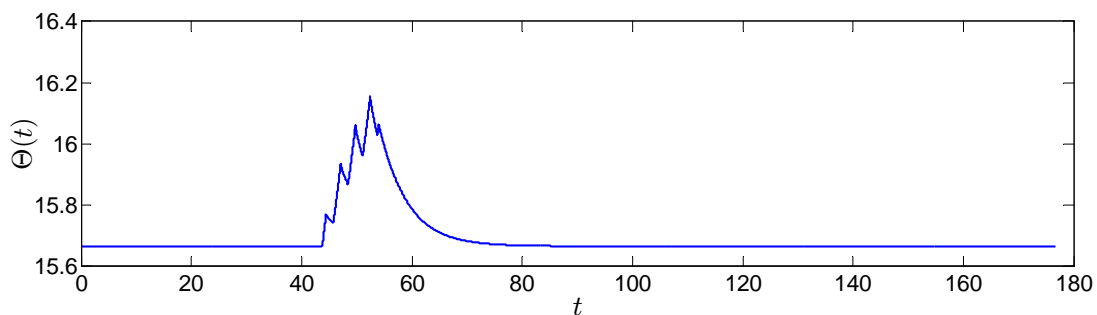
1. Zvolit konstantní délku podintervalů  $h$ , vygenerovat  $\xi_i \sim \mathcal{U}(0, 1)$ , inicializovat  $I = 0, j = 0, t_i = \frac{h}{2}$ .
  2. Pokud  $1 - \exp(-\lambda_i I) > \xi_i$ , byl nalezen odhad času  $t_i$ , ukončit algoritmus.
  3. Inkrementovat  $t_i$  o hodnotu  $h$ .
  4. Pokud  $t_i > T$ , nebyl nalezen odpovídající čas  $t_i$ , ukončit algoritmus.
  5. Inkrementovat  $I$  o hodnotu  $h \cdot b(\Theta(t_i))$ , pokračovat krokem 2.
- 

Pokud algoritmus skončí krokem 4, k plánované události by došlo až po stanovené době simulace  $T$ , událost tedy nebude naplánována.

Všimněme si, že k realizaci jednoho běhu simulace pomocí výše uvedeného algoritmu 7.1 stačí znát šest předem vygenerovat náhodných čísel z  $\mathcal{U}(0, 1)$ . Prvními třemi jsou čísla  $\xi_1, \xi_2, \xi_3$ , používaná při plánování poruch jednotlivých komponent. Další tři náhodné hodnoty  $\eta_1, \eta_2, \eta_3$ , slouží k rozhodnutí, zda po nastání poruchy komponenty bude tato komponenta zavřená ( $\eta_i < 0.5$ ) nebo otevřená ( $\eta_i \geq 0.5$ ). Pro pevně zvolené hodnoty  $\xi_1, \xi_2, \xi_3, \eta_1, \eta_2, \eta_3$  je již průběh simulace deterministický.

Výsledkem jednoho běhu simulace jsou hodnoty funkcí  $H(t)$  a  $\Theta(t)$  pro  $t \in \langle 0, t_{\max} \rangle$ , kde  $t_{\max}$  je doba poruchy systému. V případě, že během simulace (tj. do času  $T$ ) nedošlo

k poruše systému, je  $t_{\max} = T$ . Příkladem výsledku běhu simulace jsou grafy na obrázku 7.2. V tomto případě došlo v čase  $t_1 \doteq 44$  k poruše čerpadla P1 a k jeho uzavření. V časovém intervalu  $(44, 54)$  hladina vody v důsledku řídicích pravidel oscilovala mezi kritickými hodnotami 6 a 8 metrů. V čase  $t_1 \doteq 54$  došlo k poruše a otevření čerpadla P2, čímž se výška hladiny zastavila na hodnotě 7.61 metrů. V čase  $t_3 \doteq 175$  došlo k poruše ventilu V a jeho uzavření. Z důvodu poruchy všech tří komponent již nemohla být uplatněna řídicí pravidla, v čase  $t_{\max} \doteq 177$  tak došlo k přetečení.

(a) Graf funkce  $H(t)$ (b) Graf funkce  $\Theta(t)$ 

Obrázek 7.2: Příklad simulace

### 7.3 Implementace a paralelizace

Implementace byla provedena nejprve v jazyce Matlab. Algoritmus 7.2 pro naplánování poruch komponent byl zefektivněn nahrazením cyklu příkazem `cumsum`.

Následně byla provedena paralelní implementace využívající technologii CUDA. Vytvořený CUDA kernel byl volán přímo z prostředí Matlab. Poznamenejme, že paralelizace neprobíhá v rámci jednoho běhu simulace, paralelně jsou vykonávány celé samostatné běhy simulace. Pro tento způsob simulace je důležitý fakt, že jeden běh simulace je dán šesti předem vygenerovanými náhodnými čísly, v průběhu simulace není třeba generovat žádná další náhodná čísla. Všechna náhodná čísla tak mohou být předem alokována na grafické kartě jako pole náhodných čísel, tato pole jsou pak použita jako vstupní



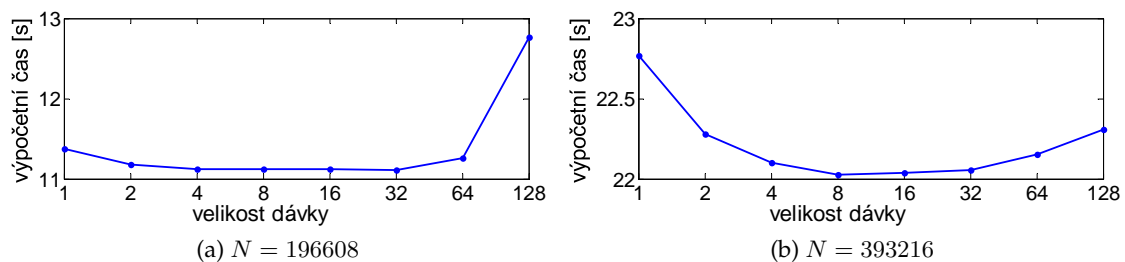
pole kernelu. Byla testována verze, v níž je v CUDA kernelu řešen pouze jeden běh simulace a verze, v níž je v CUDA kernelu řešeno více běhů simulace, tzv. dávka.

Výpočetní časy jednotlivých způsobů implementace porovnává tabulka 7.3, jsou zde uvedeny výsledky pro různé počty  $N$  prováděných běhů simulace. Jedná se vždy o průměr pěti naměřených hodnot. Ve všech případech byl při plánování poruch komponent volen diskretizační krok obdélníkového pravidla  $h = 0.1$ .

$N$	30720	101376	196608	393216
Sekvenční implementace v jazyce Matlab	263.9	852.0	-	-
Implementace pomocí CUDA (1 běh na kernel)	2.205	6.7322	11.373	22.768
Implementace pomocí CUDA (8 běhů na kernel)	1.784	5.834	11.123	22.027

Tabulka 7.3: Porovnání výpočetních časů, uvedené časy jsou v sekundách

Následující grafy ukazují závislost výpočetního času na velikosti dávky. Ve všech případech bylo použito 512 vláken na blok, počet bloků byl pak volen v závislosti na velikosti dávky tak, aby byl dosažen zvolený rozsah náhodného výběru  $N$ .



Obrázek 7.3: Závislost výpočetního času na velikosti dávky

## 7.4 Statistická analýza výsledků simulace

V případě, že náhodný pokus (běh simulace) skončí poruchou systému, je výstupem z tohoto běhu simulace čas poruchy a její typ. Pokud běh simulace neskončí poruchou, je za výstup považována informace, že do času  $T = 1000$  nedošlo k žádné z poruch.

Odhadněme nyní pomocí metody MC pravděpodobnosti, že v časovém intervalu  $\langle 0, T \rangle$  dojde k jednotlivým poruchám. Symbolem  $P_v$  označme pravděpodobnost, že do času  $T$  dojde k vysušení, symbolem  $P_p$  pravděpodobnost přetečení a symbolem  $P_t$  pravděpodobnost dosažení teploty  $100^\circ\text{C}$ .

S použitím CUDA implementace bylo provedeno  $N = 10^6$  náhodných pokusů, z toho  $N_v = 124377$  bylo ukončeno poruchou vysušení,  $N_p = 491841$  poruchou přetečení a  $N_t = 119621$  poruchou vysoká teplota. Ve zbylých  $N_0 = 264161$  pokusech k poruše nedošlo. Pro účel odhadu pravděpodobnosti vysušení považujeme za výsledky jednotlivých běhů simulace hodnoty náhodné veličiny  $Y$ , pro níž platí  $Y = 1$ , jestliže daný běh simulace skončil poruchou vysušení, a  $Y = 0$  v opačném případě. Výsledkem  $N$  běhů simulace je

tedy náhodný výběr  $Y_1, \dots, Y_N$ . Střední hodnota náhodné veličiny  $Y_v$  je pak rovna přímo hledané hodnotě  $P_v$ . Nestranným odhadem hodnoty  $P_v$  získaným pomocí metody MC je tedy výběrový průměr

$$\hat{\ell} = \frac{1}{N} \sum_{k=1}^N Y_k = \frac{N_v}{N}.$$

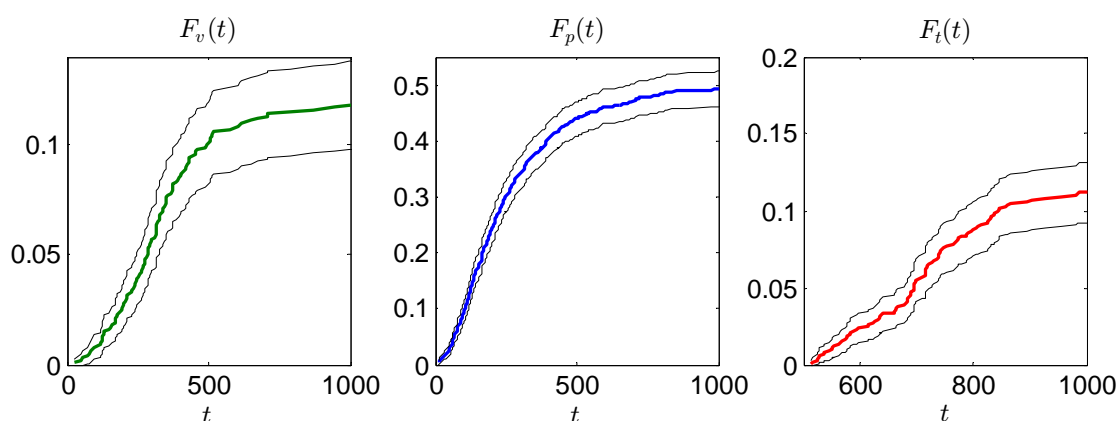
Analogicky platí, že nestranným odhadem pravděpodobnosti  $P_p$  je hodnota  $N_p \cdot N^{-1}$  a nestranným odhadem pravděpodobnosti  $P_t$  je hodnota  $N_t \cdot N^{-1}$ .

V tabulce 7.4 je uveden odhad hodnot  $P_v$ ,  $P_p$  a  $P_t$  spolu s odhady jejich přesnosti. Pro každý typ poruchy je zde uvedena výběrová směrodatná odchylka  $s$ , rozptyl  $s^2$ , relativní směrodatná odchylka, přibližný 95% interval spolehlivosti pro parametr  $\hat{\ell}$  a odhadovaný počet pokusů nutný k dosažení přesnosti  $\varepsilon = 0.01$ .

typ poruchy	$\hat{\ell}$	$s$	$s^2$	RSD	interval spolehlivosti	$N_\varepsilon$
vysušení	0.12438	0.330	0.109	0.265%	(0.12373, 0.12502)	4184
přetečení	0.49184	0.500	0.250	0.102%	(0.49086, 0.49282)	9601
vysoká teplota	0.11962	0.325	0.105	0.271%	(0.11898, 0.12026)	4046

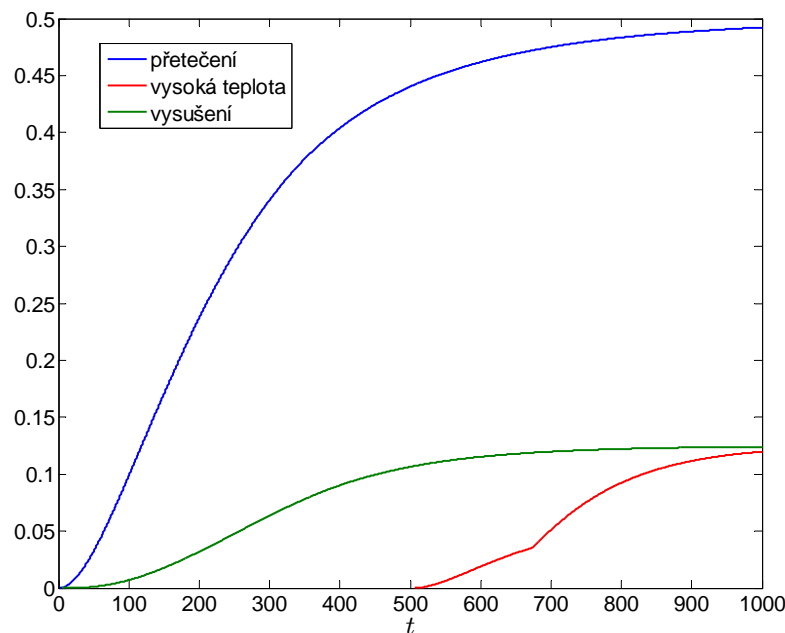
Tabulka 7.4: Odhady hodnot  $F_v(T)$ ,  $F_p(T)$  a  $F_t(T)$  pomocí metody MC

Vygenerovaný náhodný výběr v sobě však nese mnohem více informací o simulovaném systému. Stejně jako pro dobu simulace  $t = T$  lze i pro jakýkoliv jiný čas  $t \in \langle 0, T \rangle$  stanovit pravděpodobnost, že v čase kratším než  $t$  dojde k jistému typu poruchy. Pravděpodobnost, že v čase kratším než  $t$  dojde k poruše daného typu určíme jako poměr počtu poruch tohoto typu, k nimž došlo v čase kratším než  $t$ , a celkového počtu náhodných pokusů. Tímto způsobem zkonstruujeme na intervalu  $\langle 0, T \rangle$  funkce  $F_v(t)$ , resp.  $F_p(t)$ , resp.  $F_t(t)$ , které pro daný čas vrací pravděpodobnost, že v čase kratším než  $t$  dojde k poruše typu vysušení, resp. přetečení, resp. vysoká teplota. Odhadovaný průběh funkcí  $F_v(t)$ ,  $F_p(t)$  a  $F_t(t)$  pro  $t \in \langle 0, T \rangle$  ukazují grafy na obrázku 7.5.



Obrázek 7.4: Meze spolehlivosti odhadu stanovené metodou MC

Podobně jako v případě odhadů hodnot  $F_v(t)$ ,  $F_p(t)$  a  $F_t(t)$  pro  $t = T$  (viz tabulka 7.4), lze odhadnout intervaly spolehlivosti pro další časy  $t \in \langle 0, T \rangle$ . Takto získané meze, v nichž s pravděpodobností 0.95 leží funkce  $F_v(t)$ ,  $F_p(t)$  a  $F_t(t)$ , ukazují grafy na obrázku 7.4. Jsou zde znázorněny výsledky pro  $N = 10^3$ , neboť pro vysoký počet pokusů spolehlivostní meze vizuálně téměř splývají s grafem odhadované funkce.



Obrázek 7.5: Odhadovaný průběh funkcí  $F_v(t)$ ,  $F_p(t)$  a  $F_t(t)$  pro  $t \in \langle 0, T \rangle$ ;  $N = 10^6$

## 7.5 Redukce rozptylu

Za účelem snížení počtu náhodných pokusů nutných k odhadu hodnot  $P_v$ ,  $P_p$  a  $P_t$  s určitou přesností bylo aplikováno několik metod redukce rozptylu představených v kapitole 4. Konkrétně se jedná o metodu antitetických náhodných veličin, metodu řídících veličin a metodu IS s odhadem parametrů rozdělení pomocí iterační verze metody Cross-Entropy. Ve všech případech byla pro realizaci jednotlivých běhů simulace použita implementace využívající technologii CUDA.

### 7.5.1 Metoda antitetických náhodných veličin

Použití metody antitetických náhodných veličin pro řešení této úlohy odpovídá algoritmu 4.4. Náhodný výběr  $\mathbf{X}_1, \dots, \mathbf{X}_{\frac{N}{2}}$  z vícerozměrného rozdělení  $\mathcal{U}(\mathbf{0}, \mathbf{1})$  byl použit k odhadu všech tří hledaných hodnot, podobně jako tomu bylo v případě metody MC. Výsledky simulace využívající tuto metodu redukce rozptylu jsou zaznamenány v tabulce 7.5, byly opět zvoleny parametry  $N = 10^6$ ,  $\alpha = 0.05$ ,  $\varepsilon = 0.01$ . Míra redukce rozptylu pomocí této metody závisí na velikosti korelačního koeficientu náhodných veličin

$Y$  a  $Y^*$ , v tabulce je proto pro každou z odhadovaných hodnot uveden rovněž výběrový korelační koeficient  $r$ .

typ poruchy	$\hat{\ell}$	$r$	$s^2$	RSD	interval spolehlivosti	$N_\epsilon$
vysušení	0.12383	-0.141	$9.3 \cdot 10^{-2}$	0.246%	(0.12323, 0.12442)	3579
přetečení	0.49263	-0.971	$7.2 \cdot 10^{-3}$	0.017%	(0.49246, 0.49280)	279
vysoká teplota	0.11939	-0.136	$9.1 \cdot 10^{-2}$	0.253%	(0.11880, 0.11998)	3491

Tabulka 7.5: Odhady hodnot  $F_v(T)$ ,  $F_p(T)$  a  $F_t(T)$  metodou antitetických NV

Nejvýraznější redukce rozptylu byla dosažena v případě odhadu hodnoty  $P_p$ , tento výsledek lze poměrně snadno objasnit. Hledaná hodnota se vztahuje k času  $t = 1000$ , do toho času již s vysokou pravděpodobností došlo k poruchám všech systémových komponent. Jeden běh simulace je dán vstupními náhodnými čísly  $\xi_1, \xi_2, \xi_3, \eta_1, \eta_2, \eta_3$ , konfigurace systému  $(a_1, a_2, a_3)$  po nastání poruch všech komponent je však dána pouze náhodnými čísly  $\eta_1, \eta_2, \eta_3$ . Vzhledem ke konstrukci antitetických náhodných veličin je do náhodného výběru s každou trojicí  $(\eta_1, \eta_2, \eta_3)$  zahrnut rovněž nahodný pokus daný trojicí  $(1 - \eta_1, 1 - \eta_2, 1 - \eta_3)$ . Navíc jestliže pro trojici  $(\eta_1, \eta_2, \eta_3)$  přejde systém po nastání poruch všech tří komponent do stavu  $(a_1, a_2, a_3)$ , pak pro trojici  $(1 - \eta_1, 1 - \eta_2, 1 - \eta_3)$  přejde do stavu  $(1 - a_1, 1 - a_2, 1 - a_3)$ . Tabulka 7.6 uvádí závislost rychlosti přítoku  $Q_a$  na konfiguraci systému  $(a_1, a_2, a_3)$ . V případě, že již došlo k poruchám všech tří komponent, a rychlost přítoku  $Q_a$  je kladná, dojde s vysokou pravděpodobností do času  $T$  k přetečení. Nyní si již stačí všimnout, že rychlost přítoku  $Q_a$  je kladná vždy právě pro jednu z konfigurací  $(a_1, a_2, a_3)$  a  $(1 - a_1, 1 - a_2, 1 - a_3)$ , což je důvodem silné záporné korelace náhodných veličin  $Y$  a  $Y^*$ .

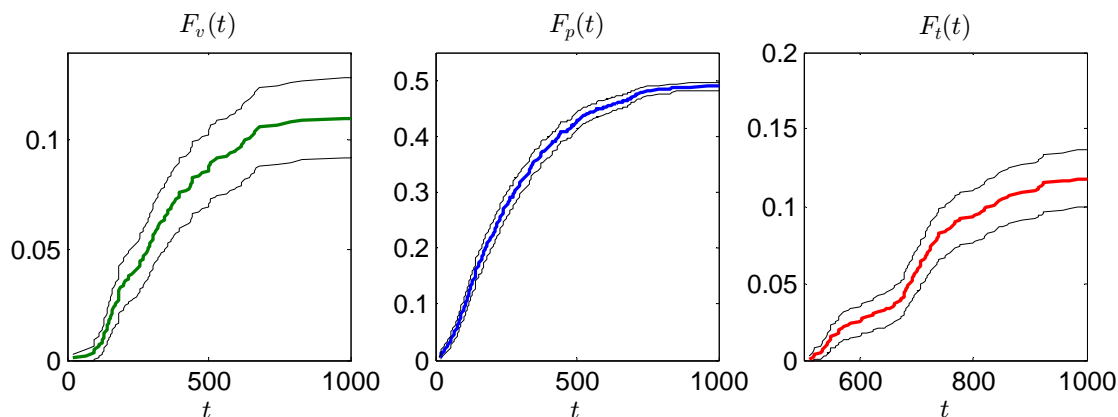
$a_1$	0	1	0	1	1	0	0	1
$a_2$	0	1	0	1	0	1	1	0
$a_3$	0	1	1	0	1	0	1	0
$Q_a$	0	1.5	-1.5	3	0	0.5	0	0.5

Tabulka 7.6: Závislost  $Q_a$  na  $a_1, a_2, a_3$

Stejně jako v případě metody MC lze pomocí metody antitetických veličin rovněž odhadnout meze, v nichž s pravděpodobností 0.95 leží funkce  $F_v(t)$ ,  $F_p(t)$  a  $F_t(t)$  pro  $t \in \langle 0, T \rangle$ , viz grafy na obrázku 7.6. Opět se jedná o výsledky pro  $N = 10^3$  náhodných pokusů.

### 7.5.2 Metoda řídicích veličin

K redukci rozptylu bude nyní použit algoritmus 4.8, který na rozdíl od algoritmu metody antitetických náhodných veličin není zcela přímočarý. Hlavním úskalím metody řídicích veličin je volba vhodné řídicí veličiny či vektoru řídicích veličin. Vhodnou řídicí veličinou je taková náhodná veličina  $C$ , která je silně korelovaná s náhodnou veličinou  $Y$ .



Obrázek 7.6: Meze spolehlivosti odhadu stanovené metodou antitetických NV

Vyjdeme-li z poznatků získaných při řešení úlohy pomocí antitetických náhodných veličin, jako možná řídicí veličina se jeví náhodná veličina rychlost přítoku po nastání všech tří poruch systému. Zvolme tedy řídicí veličinu

$$C = (I(\eta_1 < 0.5) + I(\eta_2 < 0.5) - I(\eta_3 < 0.5)) \cdot Q,$$

pro její střední hodnotu zřejmě platí

$$\ell^{(C)} = (0.5 + 0.5 - 0.5) \cdot Q = 0.75.$$

typ poruchy	$\hat{\ell}$	$r$	$s^2$	RSD	interval spolehlivosti	$N_\epsilon$
vysušení	0.12399	0.652	$6.2 \cdot 10^{-2}$	0.202%	(0.1235, 0.12448)	2405
přetečení	0.49221	-0.858	$6.6 \cdot 10^{-2}$	0.052%	(0.49171, 0.49272)	2541
vysoká teplota	0.12014	0.213	0.101	0.265%	(0.11952, 0.12077)	3881

Tabulka 7.7: Odhady hodnot  $F_v(T)$ ,  $F_p(T)$  a  $F_t(T)$  s použitím řídicí veličiny  $C$

Parametry simulace jsou opět hodnoty pro  $N = 10^6$ ,  $\alpha = 0.05$ ,  $\varepsilon = 0.01$ . Porovnání tabulek 7.4 a 7.7 ukazuje, že znatelné redukce rozptylu bylo dosaženo v případě odhadu hodnot  $P_v$  a  $P_p$ . Pro odhad všech hodnot  $P_v$ ,  $P_p$  a  $P_t$  však byla použita stejná řídicí veličina, což není optimální.

Pokusme se nyní zlepšit dosažené výsledky použitím různých řídicích veličin pro odhad každé z hodnot  $P_v$ ,  $P_p$  a  $P_t$ . Ze všech testovaných náhodných veličin vykazovaly nejlepší výsledky řídicí veličiny zkonstruované opět pomocí vstupních náhodných čísel  $\eta_1$ ,  $\eta_2$  a  $\eta_3$  s ohledem na to, v jaké konfiguraci  $(a_1, a_2, a_3)$  se systém bude nacházet po nastání poruch všech komponent. Je pravděpodobné, že dojde k vysušení, jestliže po nastání poruch všech komponent bude rychlost přítoku  $Q_a$  záporná, což podle tabulky 7.6 nastane pouze v případě, že  $(a_1, a_2, a_3) = (0, 0, 1)$ . Pro odhad hodnoty  $P_v$  proto byla použita řídicí veličina

$$C_v = I(\eta_1 < 0.5) \cdot I(\eta_2 < 0.5) \cdot I(\eta_3 > 0.5),$$

která je rovna jedné pouze v případě, že po nastání poruch všech komponent bude systém ve stavu  $(a_1, a_2, a_3) = (0, 0, 1)$ , v opačném případě bude rovna nule. Pro střední hodnotu náhodné veličiny  $C_v$  zřejmě platí

$$\ell^{(C_v)} = 0.5^3.$$

Z obdobných důvodů byla pro odhad hodnoty  $P_p$  použita řídící veličina

$$C_p = I(\eta_1 > 0.5) \cdot I(\eta_2 > 0.5) \cdot I(\eta_3 > 0.5) + I(\eta_1 > 0.5) \cdot I(\eta_2 > 0.5) \cdot I(\eta_3 < 0.5) + \\ + I(\eta_1 < 0.5) \cdot I(\eta_2 > 0.5) \cdot I(\eta_3 < 0.5) + I(\eta_1 > 0.5) \cdot I(\eta_2 < 0.5) \cdot I(\eta_3 < 0.5)$$

se střední hodnotou

$$\ell^{(C_p)} = 4 \cdot 0.5^3$$

a pro odhad hodnoty  $P_t$  řídící veličina

$$C_t = I(\eta_1 < 0.5) \cdot I(\eta_2 < 0.5) \cdot I(\eta_3 < 0.5)$$

se střední hodnotou

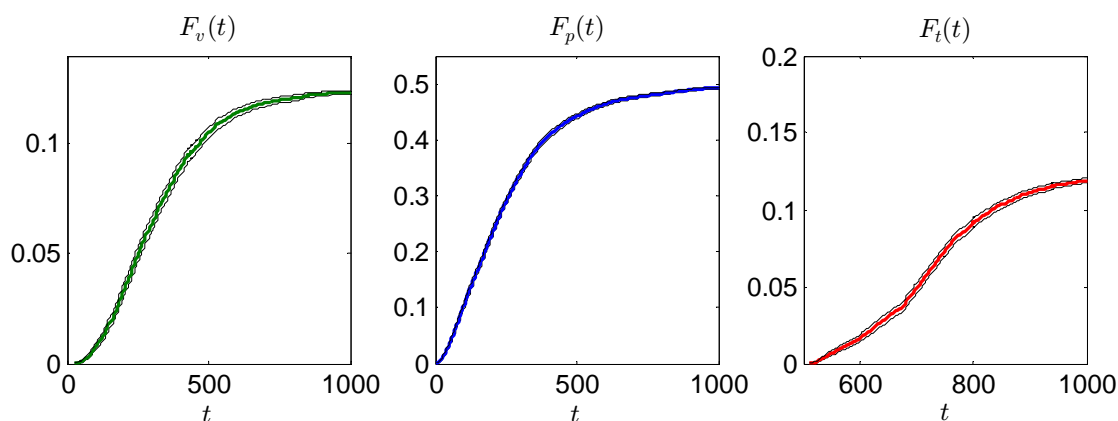
$$\ell^{(C_t)} = 0.5^3.$$

Výsledky získané pomocí řídících veličin  $C_v$ ,  $C_p$  a  $C_t$  jsou zaznamenány v tabulce 7.8.

typ poruchy	$\hat{\ell}$	$r$	$s^2$	RSD	interval spolehlivosti	$N_\varepsilon$
vysušení	0.12409	0.996	$9.1 \cdot 10^{-4}$	0.024%	(0.12403, 0.12415)	35
přetečení	0.49239	0.985	$7.5 \cdot 10^{-3}$	0.018%	(0.49222, 0.49255)	288
vysoká teplota	0.11943	0.974	$5.3 \cdot 10^{-3}$	0.061%	(0.11929, 0.11957)	204

Tabulka 7.8: Odhady hodnot  $F_v(T)$ ,  $F_p(T)$  a  $F_t(T)$  s použitím veličin  $C_v$ ,  $C_p$  a  $C_t$

Nyní již byla dosažena silná redukce rozptylu v případě všech odhadovaných hodnot. Výběrový rozptyl odhadu hodnoty  $P_v$  se snížil přibližně 120krát oproti metodě MC, odhadu hodnoty  $P_p$  přibližně 33krát a odhadu hodnotu  $P_t$  přibližně 20krát. Silná redukce rozptylu je způsobena vysokou korelací mezi původními a řídícími náhodnými veličinami, viz hodnoty výběrového korelačního koeficientu  $r$  v tabulce 7.8.



Obrázek 7.7: Meze spolehlivosti odhadu stanovené metodou řídících veličin

Pomocí metody řídicích veličin byly rovněž odhadnuty meze, v nichž s pravděpodobností 0.95 leží funkce  $F_v(t)$ ,  $F_p(t)$  a  $F_t(t)$  pro  $t \in \langle 0, T \rangle$ , viz grafy na obrázku 7.7. Grafy byly sestaveny na základě náhodného výběru o rozsahu  $N = 10^3$  za použití řídicích veličin  $C_v$ ,  $C_p$  a  $C_t$ .

### 7.5.3 Importance sampling

K redukci rozptylu bude dále použit algoritmus 4.17. Nejprve je třeba zvolit IS hustotu  $g$ , například pomocí metody Cross-Entropy, viz algoritmus 4.23.

V případě této úlohy jsou vstupní náhodná čísla  $\xi_1, \xi_2, \xi_3, \eta_1, \eta_2, \eta_3$  pro každý běh simulace generována z  $\mathcal{U}(0, 1)$ , pro použití metody CE je však nutné aby vstupní náhodná čísla pocházela z rozdělení pravděpodobnosti o hustotě  $f(\cdot, \mathbf{u})$ , tedy aby hustota pravděpodobnosti závisela na parametru  $\mathbf{u}$ . Metoda CE poté hledá IS hustotu mezi hustotami pravděpodobnosti z množiny  $\{f(\cdot, \mathbf{v}), \mathbf{v} \in \Omega\}$ . Tento problém lze vyřešit například pomocí převodu ze spojitého parametrického rozdělení pravděpodobnosti. V tomto případě bude použito vícerozměrné normální rozdělení s pevně daným rozptylem  $\sigma^2 = 1$ , optimalizovaným parametrem tedy bude střední hodnota rozdělení.

Převod z normálního rozdělení spočívá ve vygenerování náhodného vektoru  $(\xi_1^*, \xi_2^*, \xi_3^*, \eta_1^*, \eta_2^*, \eta_3^*)$  z  $\mathcal{N}(\mathbf{0}, 1)$ , tedy  $\mathbf{u} = \mathbf{0}$ , vektor  $(\xi_1, \xi_2, \xi_3, \eta_1, \eta_2, \eta_3)$  je poté získán aplikací distribuční funkce tohoto rozdělení. Díky této konstrukci pochází náhodná čísla  $\xi_1, \xi_2, \xi_3, \eta_1, \eta_2, \eta_3$  stále z  $\mathcal{U}(0, 1)$ .

Nechť je tedy původním rozdělením  $\mathcal{N}(\mathbf{0}, 1)$  o hustotě pravděpodobnosti  $f(\cdot, \mathbf{0})$ . Metoda CE byla použita zvlášť pro jednotlivé úlohy hledání  $P_v$ ,  $P_p$  a  $P_t$ , pro každou z těchto úloh tak byl nalezen optimální parametr  $\mathbf{v}$  IS hustoty  $f(\cdot, \mathbf{v})$ . Pro odhad hodnoty  $P_v$  byla stanovena IS hustota  $f(\cdot, (-0.011, 0.009, -0.007, -0.799, -0.795, 0.793))$ , pro odhad  $P_p$  IS hustota  $f(\cdot, (-0.004, 0, -0.022, 0.404, 0.407, -0.395))$  a pro odhad  $P_t$  IS hustota  $f(\cdot, (-0.040, -0.031, -0.054, -0.801, -0.797, -0.797))$ . Výsledky metody IS s těmito parametry pro  $N = 10^6$ ,  $\alpha = 0.05$  a  $\varepsilon = 0.01$  jsou uvedeny v tabulce 7.9.

typ poruchy	$\hat{\ell}$	$s^2$	RSD	interval spolehlivosti	$N_\varepsilon$
vysušení	0.12448	0.049	0.178%	(0.12405, 0.12492)	1886
přetečení	0.49313	0.194	0.089%	(0.49226, 0.49399)	7456
vysoká teplota	0.11956	0.048	0.183%	(0.11913, 0.11999)	1838

Tabulka 7.9: Odhady hodnot  $F_v(T)$ ,  $F_p(T)$  a  $F_t(T)$  metodou IS pro  $\mathcal{N}(\cdot, 1)$

Všimněme si, že první tři hodnoty vektoru  $\mathbf{v}$ , sloužící k vygenerování hodnot  $\xi_1, \xi_2, \xi_3$ , jsou ve všech případech blízké původní hodnotě (nule). Bude proto realizována ještě jedna varianta odhadu pomocí metody CE, v níž bude optimalizováno pouze rozdělení, z něhož jsou generovány hodnoty  $\eta_1, \eta_2, \eta_3$ . Úloha zůstane nezměněna, pokud budou tyto hodnoty generovány z  $\text{Ber}(0.5)$  namísto  $\mathcal{U}(0, 1)$ , neboť jsou používány pouze k porovnávání s hodnotou 0.5.

V tomto případě je tedy původním rozdělením  $\text{Ber}(\mathbf{u})$ , kde  $\mathbf{u} = \mathbf{0.5}$ . Metoda CE byla použita opět pro každou úlohu zvlášť. Pro odhad hodnoty  $P_v$  byl stanoven vek-

tor parametrů IS rozdělení  $\mathbf{v}_v = (0, 0, 1)$ , pro odhad hodnoty  $P_p$  vektor parametrů  $\mathbf{v}_p = (0.751, 0.756, 0.2534)$  a pro odhad  $P_t$  vektor parametrů  $\mathbf{v}_t = (0, 0, 0)$ . Je zřejmé, že v případě použití vektorů  $\mathbf{v}_v$  a  $\mathbf{v}_t$  se změní množina, na níž funkce  $f$  nabývá nenulových hodnot, je tedy třeba zjistit, zda IS hustoty  $f(\cdot; \mathbf{v}_v)$  a  $f(\cdot; \mathbf{v}_t)$  splňují podmínku (4.8).

Z vlastností systému plyne, že jestliže došlo k poruše vysušení, nutně musely být komponenty P1 a P2 ve stavu „zavřená a zároveň mimo provoz“ a komponenta V ve stavu „otevřená a zároveň mimo provoz“, což odpovídá vektoru  $(\eta_1, \eta_2, \eta_3) = (0, 0, 1)$ , funkce  $f(\cdot; \mathbf{v}_v)$  tedy podmínku splňuje.

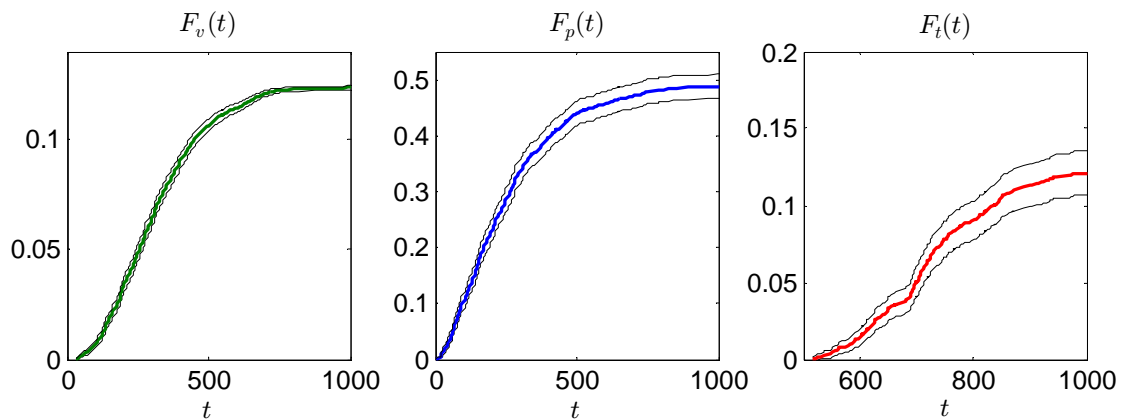
K poruše vysoká teplota však nedochází pouze v případě, že platí  $(\eta_1, \eta_2, \eta_3) = (0, 0, 0)$ . Pokud například platí  $(\eta_1, \eta_2, \eta_3) = (0, 0, 1)$ , může k poruše vysoká teplota dojít dříve, než dojde k poruše komponenty V a následnému vysušení. Funkce funkce  $f(\cdot; \mathbf{v}_t)$  tedy podmínku nesplňuje a nemůže být použita jako IS hustota.

Výsledky metody IS pro odhad hodnot  $F_v(T)$  a  $F_p(T)$  se získanými parametry  $\mathbf{v}_v$  a  $\mathbf{v}_p$  pro  $N = 10^6$ ,  $\alpha = 0.05$  a  $\varepsilon = 0.01$  jsou uvedeny v tabulce 7.10.

typ poruchy	$\hat{\ell}$	$s^2$	RSD	interval spolehlivosti	$N_\varepsilon$
vysušení	0.12416	$1.0 \cdot 10^{-4}$	0.013%	(0.12413, 0.12419)	4
přetečení	0.49306	0.121	0.112%	(0.49198, 0.49414)	4667

Tabulka 7.10: Odhady hodnot  $F_v(T)$  a  $F_p(T)$  metodou IS pro  $\text{Ber}(\mathbf{u})$

Rovněž byly odhadnuty meze, v nichž s pravděpodobností 0.95 leží funkce  $F_v(t)$ ,  $F_p(t)$  a  $F_t(t)$  pro  $t \in \langle 0, T \rangle$ , viz grafy na obrázku 7.8. Grafy byly sestaveny na základě náhodného výběru o rozsahu  $N = 10^3$  pomocí metody IS, v případě odhadu průběhu funkcí  $F_v(t)$  a  $F_p(t)$  bylo použito Bernoulliho rozdělení a v případě funkce  $F_t(t)$  normální rozdělení.



Obrázek 7.8: Meze spolehlivosti odhadu stanovené metodou IS

Získané odhady průběhu funkcí  $F_v(t)$ ,  $F_p(t)$  a  $F_t(t)$  odpovídají grafům uvedeným v [11], kde byla řešena identická úloha.



#### 7.5.4 Porovnání jednotlivých metod

Hlavním kritériem efektivity metod použitých v této sekci je dosažené snížení rozptylu oproti základní metodě MC. Následující tabulka proto uvádí odhadovanou redukci rozptylu dosaženou při odhadu hodnot  $F_v(T)$ ,  $F_p(T)$  a  $F_t(T)$  pomocí jednotlivých metod. V případě metody IS je uveden výběrový rozptyl odhadů hodnot  $F_v(T)$  a  $F_p(T)$  získaný při použití Bernoulliho rozdělení a výběrový rozptyl odhadu hodnoty  $F_t(T)$  získaný při použití normálního rozdělení.

Metoda	$F_v(T)$ - vysušení		$F_p(T)$ - přetečení		$F_t(T)$ - vysoká teplota	
	$s^2$	redukce	$s^2$	redukce	$s^2$	redukce
MC	0.109	1×	0.250	1×	0.105	1×
Antitické NV	0.093	1.2×	$7.2 \cdot 10^{-3}$	35×	0.091	1.15×
Řídící veličiny	$9.1 \cdot 10^{-4}$	119×	$7.5 \cdot 10^{-3}$	33×	$5.3 \cdot 10^{-3}$	20×
IS	$1.0 \cdot 10^{-4}$	1090×	0.121	2.1×	0.048	2.2×

Tabulka 7.11: Porovnání metod redukce rozptylu

## 8 Aplikace II - spolehlivost systémů s nezávislými prvky

Tato sekce se zabývá obecným problémem kvantifikace spolehlivosti systémů s nezávislými prvky. Na základě znalosti pravděpodobnosti provozu jednotlivých prvků, z nichž je systém složen, je určována pravděpodobnost, že je v provozu celý systém.

Se spolehlivostí souvisí také pojem pohotovost. Pohotovost dynamického systému v čase  $t$  je definována jako pravděpodobnost, že je systém v tomto čase v provozu. Jsou-li známy pravděpodobnosti provozu jednotlivých komponent v čase  $t$ , je možné se omezit pouze na zachycení systému v čase  $t$  a chápat tak systém jako statický. V těchto případech je v literatuře pojem pohotovost obvykle nahrazován obecnějším pojmem spolehlivost.

Model komponenty je často vyjadřován pomocí náhodných veličin doba do poruchy a doba do opravy pocházejících z exponenciálního rozdělení. Je-li komponenta v čase  $t = 0$  v provozu, je pravděpodobnost provozu komponenty v čase  $t$  dána funkcí

$$p(t) = \frac{\mu}{\mu + \lambda} + \frac{\lambda}{\mu + \lambda} e^{-(\mu + \lambda)t},$$

kde  $\lambda$  je parametrem doby do poruchy, tedy tzv. intenzitou poruch, a  $\mu$  je parametrem doby do opravy, tzv. intenzitou oprav, viz například [5]. Limitním přechodem  $t \rightarrow \infty$  je získán tzv. asymptotický koeficient pohotovosti  $p_\infty = \mu / (\mu + \lambda)$ . V [15] jsou dále uvedeny funkce  $p(t)$  a příslušné koeficienty  $p_\infty$  pro případ, kdy komponenta podléhá dvěma typům poruch a případ, kdy náhodná veličina doba do poruchy pochází z Weibullova či logaritmicko-normálního rozdělení pravděpodobnosti. Při studiu pohotovosti systému má své opodstatnění dosadit jako pravděpodobnost provozu komponenty jak hodnotu  $p(t)$  pro zvolenou hodnotu  $t$ , tak i asymptotický koeficient  $p_\infty$ . Dále bude pravděpodobnost provozu jednotlivých komponent systému chápána jako součást zadání úlohy.

Reálné systémy tohoto typu obvykle pracují s vysoce spolehlivými prvky, úloha tak vede na problém kvantifikace řídkých jevů. Pro odhad spolehlivosti bude použita metoda IS s několika variantami odhadu parametrů IS rozdělení. Dále bude na řešení problém aplikována metoda PMC (z anglického Permutation Monte Carlo) představená v [2]. Vedle odhadu spolehlivosti bude provedena také analýza citlivosti testovaných systémů. Použité metody budou pro zvýšení efektivity paralelizovány s použitím technologie CUDA.

### 8.1 Specifikace systému

Dále bude uvažován statický systém, který je určen množinou  $Q = \{a, c_1, \dots, c_n, b\}$  a relací  $R$  na množině  $Q$ . Prvek  $a$  je vstupním prvkem systému,  $b$  je výstupním prvkem systému a  $C = \{c_1, \dots, c_n\}$  je množinou všech komponent. O každých dvou prvcích množiny  $Q$  je možné rozhodnout, zda jsou, či nejsou v relaci. Skutečnost, že prvek  $q_i \in Q$  je v relaci s prvkem  $q_j \in Q$ , je značena zápisem  $(q_i, q_j) \in R$ . Obecně se nejedná o symetrickou relaci, tedy pokud platí  $(q_i, q_j) \in R$ , nemusí platit  $(q_j, q_i) \in R$ . [16]

Stochastické vlastnosti systému jsou dány náhodným vektorem  $B = (B_1, \dots, B_n)$ . Komponentě  $c_i$  je přiřazena náhodná veličina  $B_i$  pocházející z Bernoulliho rozdělení s pa-

rametrem  $p_i$ , tedy  $\mathbb{P}(B_i = 1) = p_i$ . Jev  $B_i = 0$  indikuje poruchu dané komponenty, zatímco jev  $B_i = 1$  značí, že je tato komponenta v provozu.

Systém je v provozu, jestliže existuje taková posloupnost indexů funkčních komponent  $k_1, \dots, k_d$ , kde  $d \in \mathbb{N}$ , že platí

$$(a, c_{k_1}) \in R, (c_{k_d}, b) \in R \text{ a } \forall i \in \{1, \dots, d-1\} : (c_{k_i}, c_{k_{i+1}}) \in R.$$

Spolehlivost systému je definována jako pravděpodobnost, že je systém v provozu. Úlohu je však vhodnější formulovat jako problém hledání nespolehlivosti systému, tedy pravděpodobnosti  $\ell$ , že systém není v provozu. Pro nespolehlivost systému zřejmě platí

$$\ell = \mathbb{E}(H(\mathbf{B})),$$

kde  $H(\mathbf{b}) = 1$ , jestliže pro stavový vektor  $\mathbf{b}$  je systém mimo provoz, a  $H(\mathbf{b}) = 0$  v opačném případě. Implementaci funkce  $H$  se věnuje sekce 8.4.3.

Takto specifikovaný systém je zobecněním více typů systémů, kterými se zabývá dostupná literatura. Jedná se například o problém kvantifikace spolehlivosti sítě, viz [2], nebo o řešení pohotovosti systémů, viz [5].

**Úloha 8.1** (Spolehlivost sítě, viz [2]). Sítí je myšlen graf  $G = (V, E)$ , kde

- $V$  značí množinu všech vrcholů,
- $E$  značí množinu všech hran, jedná se tedy o množinu dvouprvkových podmnožin množiny  $V$ , platí  $|E| = n$ . [17]

Navíc jsou specifikovány dva koncové vrcholy  $a \in V$  a  $b \in V$ , vrchol  $a$  je zároveň vstupním prvkem a vrchol  $b$  prvkem výstupním. Komponentami systému jsou všechny hrany grafu  $G$ , tedy  $C = E$ . Dvě hrany jsou v relaci, jestliže mají společný vrchol, hrana a koncový vrchol jsou v relaci, jestliže tento vrchol náleží dané hraně. Jelikož se nejedná o orientované hrany, je relace  $R$  symetrická.

Spolehlivostí sítě je zde pravděpodobnost, že jsou koncové vrcholy  $a$  a  $b$  propojeny funkčními hranami.

**Úloha 8.2** (Pohotovost systému, viz [5]). Systémem je myšlen orientovaný graf  $G = (V, E)$ , kde

- $V$  značí množinu všech vrcholů, platí  $|V| = n + 2$ ,
- $E$  značí množinu všech hran, jedná se tedy o množinu uspořádaných dvojic prvků množiny  $V$ .

Navíc je specifikován vstupní prvek  $a \in V$  a výstupní prvek  $b \in V$ . Komponentami systému jsou prvky množiny  $C = V \setminus \{a, b\}$ . Vrchol  $q_i$  je v relaci s vrcholem  $q_j$ , jestliže v vrcholu  $q_i$  vede orientovaná hrana do vrcholu  $q_j$ , tj.  $(q_i, q_j) \in R \Leftrightarrow (q_i, q_j) \in E$ .

Pohotovost systému je rovna pravděpodobnosti, že existuje orientovaná cesta vedoucí z vrcholu  $a$  do vrcholu  $b$  pouze přes funkční komponenty.

## 8.2 Dynamická formulace

Popsaný statický systém lze chápat také jako alternativní dynamický systém zachycený v konkrétním časovém okamžiku. Alternativní dynamický systém je rovněž tvořen množinou  $Q$  a relací  $R$ , v tomto případě se však stav jednotlivých komponent v čase mění. V čase  $t = 0$  jsou všechny komponenty  $c_i \in C$  mimo provoz. Náhodná veličina  $X_i$  udává dobu do opravy  $i$ -té komponenty. Rozdělení pravděpodobnosti náhodného vektoru  $\mathbf{X} = (X_1, \dots, X_n)$  je stanoveno tak, aby pro jistou pevně zvolenou hodnotu  $T \in \mathbb{R}$  platilo

$$\mathbb{P}(X_i < T) = \mathbb{P}(B_i = 1) = p_i. \quad (8.1)$$

Stav systému v čase  $t$  je značen  $\mathbf{B}(t) = (B_1(t), \dots, B_n(t))$ , čímž je definován náhodný proces  $\{\mathbf{B}(t), t \geq 0\}$ . Platí  $B_i(t) = 1$ , jestliže je komponenta  $c_i$  v čase  $t$  v provozu, v opačném případě  $B_i(t) = 0$ .

S využitím dynamické formulace lze hodnotu  $\ell$  zapsat jako

$$\ell = \mathbb{E}(\mathbf{I}_{S(\mathbf{X}) \geq T}) = \mathbb{P}(S(\mathbf{X}) \geq T),$$

kde  $S(\mathbf{x})$  udává pro konkrétní vektor  $\mathbf{x}$  dobu, kdy bude systém uveden do provozu.

## 8.3 Použité metody redukce rozptylu

Pro redukci rozptylu odhadu nespolehlivosti  $\ell$  bude použita metoda IS, aplikovaná na statickou i dynamickou formulaci systému. Bude popsána metoda SPB (z anglického *single parameter biasing*), která je v [5] použita pro řešení pohotovosti systémů, viz úloha 8.2, zde je chápána jako metoda volby parametrů IS rozdělení. Rovněž bude navrhnut efektivnější způsob odhadu parametrů založený na metodě CE, která automaticky zohledňuje strukturu systému.

### 8.3.1 Metoda IS

Náhodný vektor  $\mathbf{B}$  pochází z  $n$ -rozměrného Bernoulliho rozdělení daného vektorem parametrů  $\mathbf{p} = (p_1, \dots, p_n)$ , IS rozdělení je voleno ze stejné pravděpodobnostní rodiny. K odhadu parametrů IS rozdělení je možné použít základní iterační algoritmus metody CE, viz 4.23, jehož výsledkem je vektor  $\boldsymbol{\eta} = (\eta_1, \dots, \eta_n)$ , který je použit jako parametr IS rozdělení.

Při testování docházelo k nastavení některého z parametrů  $\eta_i$ , kde  $i \in \{1, \dots, n\}$ , na hodnotu 1, výsledné IS rozdělení by tak nesplňovalo podmínku (4.8). Tento problém lze vyřešit použitím tzv. *screening* algoritmu, viz [1, str. 155], nebo pomocí vyhlazovacího parametru, viz 4.24. Zde se osvědčila první z těchto možností, byl tedy použit algoritmus 8.3 kombinující CE a *screening* algoritmus. Nové IS rozdělení má upřednostnit vzorky, pro něž nastane porucha systému, není tedy žádoucí, aby platilo  $\eta_i > p_i$ ,  $i \in \{1, \dots, n\}$ , což by způsobilo snížení pravděpodobnosti poruchy  $i$ -té komponenty. Této situaci zamezí bod 5. následujícího algoritmu, pokud by některá z hodnot  $\eta_i$ ,  $i \in \{1, \dots, n\}$  měla být nastavena na hodnotu vyšší než  $p_i$ , příslušná komponenta je označena za nepodstatnou a daný parametr není aktualizován.

---

**Algoritmus 8.3** (Metoda CE pro kvantifikaci spolehlivosti sítě). Vstupem algoritmu je počet iterací  $k_{\max}$ .

1. Zvolit počáteční vektor parametrů  $\boldsymbol{\eta}^{(0)}$ , například  $\boldsymbol{\eta}^{(0)} = \boldsymbol{p}$ , nastavit  $k = 1$  a  $I = \{1, \dots, n\}$ .
2. Vygenerovat náhodný výběr  $\boldsymbol{B}_1, \dots, \boldsymbol{B}_M$  z  $n$ -rozměrného Bernoulliho rozdělení s nezávislými složkami daného vektorem parametrů  $\boldsymbol{\eta}^{(k-1)}$ .
3. Pokud  $H(\boldsymbol{B}_1) = \dots = H(\boldsymbol{B}_M)$ , opakovat krok 2.

4. Určit

$$\boldsymbol{\eta}^{(k)} = \underset{\boldsymbol{\eta}}{\operatorname{argmax}} \frac{1}{N} \sum_{i=1}^N H(\boldsymbol{B}_i) W(\boldsymbol{B}_i; \boldsymbol{\theta}, \boldsymbol{\eta}^{(k-1)}) \ln f(\boldsymbol{B}_i; \boldsymbol{\eta}),$$

5. Ověřit  $\forall i \in I$ , zda  $\eta_i < p_i$ , pokud ne, odebrat  $i$  z množiny  $I$  a nastavit  $\eta_i = p_i$ .
  6. Je-li  $k < k_{\max}$ , navýšit iterační proměnnou  $k$  o 1 a pokračovat krokem 2.
  7. Pro kvantifikaci nespolehlivosti systému použít algoritmus 4.17, jako IS rozdělení použít  $n$ -rozměrné Bernoulliho rozdělení s vektorem parametrů  $\boldsymbol{\eta}^{(k)}$ , jako množinu důležitých prvků použít výslednou množinu  $I$ .
- 

V případě řídkých jevů může nízká pravděpodobnost vygenerování vzorků  $\boldsymbol{B}_i$ , pro které  $H(\boldsymbol{B}_i) = 1$ , způsobit problém popsáný v sekci 4.4.4. Tento problém částečně řeší krok 3. algoritmu 8.3, účinněji lze situaci zabránit vhodnou volbou počátečního vektoru parametrů  $\boldsymbol{\eta}^{(0)}$ , viz příklady 8.6 a 8.7.

V [5] je pro redukci rozptylu použita metoda SPB. Jedná se v podstatě o metodu IS, kterou lze použít, pokud jednotlivé složky stavového vektoru pochází z Bernoulliho rozdělení. Jako IS rozdělení je použito  $n$ -rozměrné Bernoulliho rozdělení dané vektorem parametrů  $\boldsymbol{\eta} = \alpha \cdot \boldsymbol{p}$ , kde  $\alpha \in (0, 1)$ . Díky této volbě IS hustoty se zvýší pravděpodobnost poruch jednotlivých komponent a tím i pravděpodobnost poruchy systému. Na rozdíl od předchozího způsobu volby parametrů však není zohledněna důležitost jednotlivých komponent.

Statická formulace systému nepracuje s úrovní  $\gamma$ , nelze tak použít algoritmus 4.26 metody CE pro kvantifikaci řídkých jevů. Nabízí se však využití dynamické formulace úlohy s volbou  $\gamma = T$ . V příkladech v sekci 8.5 byla testována i tato varianta metody IS, byl zvolen náhodný vektor  $\boldsymbol{X}$  pocházející z  $n$ -rozměrného exponenciálního rozdělení a  $T = 1$ . K určitému zvýšení efektivity vedlo použití vyhlazovacího parametru, viz 4.24.

### 8.3.2 Metoda PMC

Metoda PMC je speciální variantou podmíněné metody MC, v [2] je použita pro kvantifikaci spolehlivosti sítě, viz úloha 8.1, princip metody lze však využít i při řešení obecné úlohy dané množinou  $Q$  a relací  $R$ .

Tato metoda redukce rozptylu využívá dynamickou formulaci systému popsanou v předchozí sekci. Je zvoleno  $T = 1$  a jednotlivé složky náhodného vektoru  $\mathbf{X}$  pochází z exponenciálního rozdělení s parametrem  $\lambda_i$ . Lze snadno odvodit, že pro splnění rovnosti (8.1) je třeba volit

$$\lambda_i = \frac{1}{T} (-\ln(1 - p_i)) = -\ln(1 - p_i).$$

Označme nyní  $\Pi = (\Pi_1, \dots, \Pi_n)$  permutaci čísel  $1, \dots, n$  danou seřazením hodnot  $X_1, \dots, X_n$  od nejnižší po nejvyšší, tedy  $X_{\Pi_1} < \dots < X_{\Pi_n}$ . Jelikož  $X_1, \dots, X_n$  značí doby do opravy jednotlivých komponent, náhodný vektor  $\Pi$  udává pořadí, v němž byly jednotlivé komponenty uvedeny do provozu. Náhodná veličina  $A_j$  značí dobu, po kterou proces  $\{\mathbf{B}(t), t \geq 0\}$  setrval v  $j$ -tém nabytém stavu, tedy  $A_1 = X_{\Pi_1}$ ,  $A_2 = X_{\Pi_2} - X_{\Pi_1}$ , apod. Funkce  $\text{crit}(\Pi)$  udává tzv. kritickou hodnotu, tedy počet komponent, které bylo nutné uvést do provozu, aby byl v provozu celý systém, platí tedy

$$S(\mathbf{X}) = X_{\text{crit}(\Pi)} = A_1 + \dots + A_{\text{crit}(\Pi)}.$$

Princip podmíněné metody MC pro odhad střední hodnoty náhodné veličiny  $Y$  spočívá ve volbě takového náhodného vektoru  $\mathbf{Z}$ , že  $\mathbb{E}(Y|\mathbf{Z})$  lze snadno určit. V tomto případě  $Y = I_{S(\mathbf{X}) \geq 1}$  a  $\mathbf{Z} = \Pi$ , hledaná pravděpodobnost  $\ell$  je tedy odhadována hodnotou

$$\hat{\ell} = \frac{1}{N} \sum_{k=1}^N \mathbb{E}(I_{S(\mathbf{X}) \geq 1} | \Pi_k),$$

kde  $\Pi_1, \dots, \Pi_N$  je náhodný výběr jednoznačně určený náhodným výběrem  $\mathbf{X}_1, \dots, \mathbf{X}_N$  z  $n$ -rozměrného exponenciálního rozdělení s nezávislými složkami daného vektorem parametrů  $\lambda = (\lambda_1, \dots, \lambda_n)$ . Pro další postup je tedy stěžejní analytický výpočet hodnoty

$$G(\pi) = \mathbb{E}(I_{S(\mathbf{X}) \geq 1} | \Pi = \pi) = \mathbb{P}(A_1 + \dots + A_{\text{crit}(\pi)} \geq 1).$$

V [2] je využit poznatek, že  $\{\mathbf{B}(t), t \geq 0\}$  je Markovův proces, a hodnota  $G(\pi)$  je počítána jako

$$G(\pi) = \mathbb{P}(A_1 + \dots + A_{\text{crit}(\pi)} \geq 1) = \sum_{j=i}^b \omega_{c,j} \exp(-\nu_{b-j+1}),$$

kde  $c = \text{crit}(\pi)$ ,

$$\nu_i = \sum_{j=1}^i \lambda_{\Pi_j} \tag{8.2}$$

pro  $i \in \{1, \dots, c\}$ , a hodnoty  $\omega_{c,j}$  jsou dány rekursivním předpisem

$$\omega_{1,1} = 1, \quad \omega_{k+1,j} = \omega_{k,j} \frac{\nu_{c-k}}{\nu_{c-k} - \nu_{c-j+1}}, \quad \omega_{k+1,k+1} = 1 - \sum_{j=1}^k \omega_{k+1,j}$$

pro  $k \in \{1, \dots, c-1\}$  a  $j \in \{1, \dots, k\}$ .

## 8.4 Implementace a paralelizace

Časově nejnáročnější algoritmy používané při kvantifikaci spolehlivosti sítě byly paralelizovány s využitím technologie CUDA. Jedná se o funkci  $G(\pi)$  používanou při redukci rozptylu pomocí metody PMC, funkci  $S(x)$  používanou navíc při redukci rozptylu pomocí metody IS aplikované na dynamickou formulaci systému a funkci  $H(b)$  používanou pouze při aplikaci metody IS na statickou formulaci systému.

### 8.4.1 Funkce $G(\pi)$

V [2, str. 556] lze nalézt sekvenční kód v jazyce Matlab pro výpočet hodnoty  $G(\pi)$ . V této implementaci jsou hodnoty  $\nu_k$ , kde  $k \in \{1, \dots, b\}$ , vypočteny předem a následně použity při výpočtu hodnot  $\omega_{k,j}$ , kde  $k \in \{1, \dots, b\}$  a  $j \in \{1, \dots, k\}$ , a zaznamenány do matice  $W$ . Tento přístup není vhodný pro paralelizaci pomocí CUDA, neboť jednotlivá vlákna mají k dispozici pouze malé množství lokální paměti.

Pro výpočet hodnot  $\omega_{b,j}$ , kde  $j \in \{1, \dots, b\}$ , lze však odvodit explicitní vzorec

$$\omega_{b,j} = \prod_{i=1, i \neq k-1}^b \frac{\nu_i}{\nu_i - \nu_{b-k+1}}. \quad (8.3)$$

Není tedy potřeba zaznamenávat matici  $W$ , což vede ke snížení paměťových nároků implementace. Rovněž hodnoty  $\nu_i$  není nutné počítat předem, je možné vždy použít předpis (8.2) a znovu tak snížit paměťové nároky, testování však ukázalo, že průběžné výpočty hodnot  $\nu_i$  výrazně zvyšují výpočetní čas.

Během testování bylo dále zjištěno, že v případě nízkých hodnot pravděpodobností  $p_i$  je tento algoritmus velice náchylný na přesnost výpočtu. Jelikož při CUDA implementaci na běžných grafických kartách nejsou výpočty s dvojnásobnou přesností efektivní, byla pro úlohy s nízkými hodnotami  $p_i$  implementována rovněž CPU verze algoritmu využívající explicitní vzorec (8.3) a paralelní cyklus `parfor`. Nicméně v případě vysoce spolehlivých systémů obvykle platí pro pravděpodobnosti provozu jednotlivých komponent  $p_i > 0.9$ , jednoduchá přesnost používaná v CUDA je tedy ve většině případů postačující. Tabulka 8.1 porovnává výpočetní čas jednotlivých implementací výpočtu hodnoty  $G(\pi)$ . Konkrétně je jedná o původní sekvenční kód a o tři varianty vlastní implementace využívající vzorec (8.3), sekvenční algoritmus, verzi využívající cyklus `parfor` a GPU verzi algoritmu. Uvedené časy byly naměřeny při řešení úloh 8.6 a 8.7, jež budou uvedeny dále, pro  $p_i = 1 - 10^{-7}$  a  $10^5$  náhodných pokusů. Jedná se pouze o rozdílné způsoby implementace téže metody, výsledné hodnoty  $G(\pi)$  jsou tedy ve všech případech stejné.

typ implementace	paralelizace	18 komponent	60 komponent
viz [2, str. 556]	-	24.531 s	261.688 s
vlastní implementace využívající vzorec (8.3)	-	1.107 s	5.179 s
	parfor	1.097 s	3.072 s
	CUDA	0.098 s	0.634 s

Tabulka 8.1: Výpočetní čas jednotlivých implementací funkce  $G(\pi)$

Zdrojový kód funkce  $G(\pi)$  v CUDA je uveden v příloze, viz B.3.

#### 8.4.2 Počítačová reprezentace systému

V literatuře, například [18], je diskutováno několik způsobů konstrukce výkonnostní funkce  $H(b)$  pro speciální případy systémů, zde bude použit obecný způsob založený na reprezentaci relace  $R$  pomocí matice.

Relaci  $R$  na množině  $Q$  je možné reprezentovat pomocí matice  $M$  logických hodnot 0 a 1 o rozměrech  $(n+2) \times (n+2)$ . Hodnota 1 na pozici  $(i, j)$  značí, že  $(p_i, p_j) \in R$ . První řádek a sloupec odpovídá vstupnímu prvku  $a$ , poslední řádek a sloupec výstupnímu prvku  $b$ , zbylé řádky a sloupce postupně komponentám  $c_1, \dots, c_n$ .

V případě úlohy 8.2 je  $M$  maticí sousednosti grafu  $G$ , hodnota 1 na pozici  $(i, j)$  značí, že z vrcholu  $p_i$  vede orientovaná hrana do vrcholu  $p_j$ .

#### 8.4.3 Funkce $H(b)$

Funkce  $H(b)$  vrací hodnotu 1, je-li pro daný stavový vektor  $b$  systém mimo provoz, a hodnotu 0 v opačném případě. K vyhodnocení funkce  $H$  byl použit algoritmus uvedený v [19], vytvořený původně pro řešení úlohy 8.2, lze však použít i k řešení problému zadaného množinou  $Q$  a relací  $R$ .

---

##### Algoritmus 8.4 (Vyhodnocení funkce $H(b)$ ).

Vstupem algoritmu je vektor  $b$  a matice  $M$ .

1. Vytvořit prázdnou množinu  $M_1$ , množinu  $M_2$  obsahující funkční komponenty  $c_i$  takové, že  $(a, c_i) \in R$  množinu  $M_3$  obsahující nefunkční komponenty.
  2. Je-li  $M_2$  prázdná, systém není v provozu,  $H(b) = 1$ . Ukončit algoritmus.
  3. Pokud  $M_2$  obsahuje takovou komponentu  $c_i$ , že  $(c_i, b) \in R$ , systém je v provozu,  $H(b) = 0$ . Ukončit algoritmus.
  4. Zapsat do množiny  $M_1$  všechny komponenty  $c_i$ , které nejsou v množině  $M_3$ , a zároveň  $\exists q \in M_2 : (q, c_i) \in R$ .
  5. Přesunout prvky množiny  $M_2$  do  $M_3$ . Přesunout prvky množiny  $M_1$  do  $M_2$ .
  6. Pokračovat krokem 2.
- 

Každou z množin  $M_1$ ,  $M_2$  a  $M_3$  lze chápat jako vektor hodnot 0 a 1 o délce  $n$ . Hodnota 1 na pozici  $i$  značí, že  $i$ -tá komponenta na do množiny patří, a naopak. Z důvodu snížení paměťových nároků CUDA implementace jsou používány bitové operace, což vede rovněž ke zkrácení výpočetního času. Matici  $M$  i množiny  $M_1$ ,  $M_2$  a  $M_3$  reprezentuje pole proměnných datového typu `unsigned int`, z nichž každá má délku 32 bitů. Jednotlivé bity v případě množin  $M_1$ ,  $M_2$  a  $M_3$  určují přítomnost komponenty v dané množině a v případě matice  $M$  relaci mezi jednotlivými hranami a koncovými vrcholy. Zdrojový kód této funkce je uveden v příloze, viz B.4.



#### 8.4.4 Funkce $S(x)$

Tato funkce pro daný vektor časových hodnot  $x$  určí dobu  $S(x)$ , kdy bude systém uveden do provozu. V [2, str. 553] je uveden sekvenční kód v jazyce Matlab, používaný při řešení úlohy 8.1. V průběhu výpočtu je na základě pořadí uvádění komponent do provozu postupně konstruována matice incidence mezi jednotlivými hranami a vrcholy a je rozhodováno, zda je daný graf průchozí.

Při paralelizaci pomocí CUDA není z paměťových důvodů vhodné použít tuto implementaci. Průběh konstrukce matice incidence se pro jednotlivé pokusy liší a pro každé vlákno by tak musela být zaznamenávána vlastní matice incidence. Byla tedy zvolena implementace vycházející z algoritmu 8.4 pro vyhodnocení funkce  $H$ , jejíž použití je navíc obecnější. Nová implementace pracuje s maticí  $M$ , která je pro všechna vlákna společná. Matici  $M$  je možné pro snížení výpočetního času uložit do konstantní paměti.

Daný vektor  $x$  je seřazen na CPU a na základě tohoto seřazení je určena permutace  $\pi$ , pomocí algoritmu 8.5 je na GPU nalezena kritická hodnota  $\text{crit}(\pi)$  a doba uvedení systému do provozu je následně určena jako  $x_{\text{crit}(\pi)}$ .

---

#### Algoritmus 8.5 (Vyhodnocení funkce $\text{crit}(\pi)$ ).

Vstupem algoritmu je permutace  $\pi$  a matice  $M$ .

1. Inicializovat nulový stavový vektor  $b$  o délce  $n$ . Inicializovat proměnnou  $k = 1$ .
  2. Zapsat číslo 1 na pozici  $\pi_k$  vektoru  $b$ .
  3. Vyhodnotit  $H(b)$  pomocí algoritmu 8.4.
  4. Je-li systém v provozu, nastavit  $\text{crit}(\pi) = k$  a ukončit algoritmus.
  5. Navýšit hodnotu proměnné  $k$  o 1 a pokračovat krokem 2.
- 

### 8.5 Test použitých algoritmů

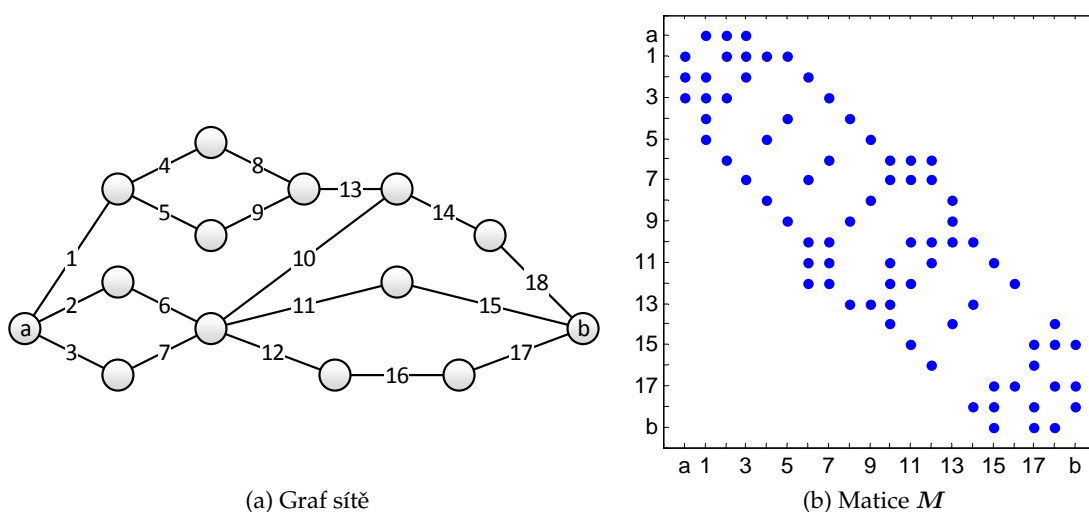
Algoritmy představené v předchozí sekci budou nyní aplikovány na několik konkrétních úloh. Jedná se o následující algoritmy:

1. Metoda IS použitá pro řešení statické formulace systému dané Bernoulliho rozdělením s odhadem parametrů IS hustoty pomocí metody CE kombinované se *screening* algoritmem, viz algoritmus 8.3. Funkce  $H(b)$  je paralelizována pomocí CUDA.
2. Metoda IS použitá pro řešení dynamické formulace systému dané exponenciálním rozdělením s odhadem parametrů IS hustoty pomocí metody CE pro kvantifikaci řídkých jevů, viz algoritmus 4.26. Funkce  $S(x)$  je paralelizována pomocí CUDA.
3. Metoda PMC:
  - (a) sekvenční implementace uvedená v [2],

- (b) metoda PMC paralelizovaná pomocí CUDA,
- (c) hybridní verze, kde funkce  $S(x)$  je paralelizována pomocí CUDA a funkce  $G(\pi)$  pomocí cyklu `parfor`.

Pro srovnání bude na některé úlohy aplikována také metoda SPB a základní metoda MC.

**Příklad 8.6** (Testovací systém převzatý z [2, str. 558]). Jedná se o úlohu typu 8.1, systémem je síť tvořená 18 komponentami, viz obrázek 8.1. Pravděpodobnost provozu jednotlivých komponent je shodná, tj.  $p_i = p$ ,  $i \in \{1, \dots, 18\}$ , úloha byla řešena pro více různých hodnot  $p = 1 - 0.1^m$ , kde  $m \in \{1, 2, \dots, 10\}$ .



Obrázek 8.1: Znázornění systému z příkladu 8.6

V případě všech testovaných algoritmů bylo provedeno  $N = 10^6$  náhodných pokusů.

1. První varianta metody IS navíc zahrnovala  $M = 10^5$  pokusů metody CE v každé iteraci, přičemž iterací bylo vykonáno  $k_{\max} = 10$ . Jelikož byla stejná úloha řešena pro více různých hodnot  $p$ , kterými byla postupně zvyšována spolehlivost systému, byl pro  $m > 1$  jako počáteční vektor  $\eta^{(0)}$  použit vždy vektor  $\eta^{(k_{\max})}$  spočtený pro předchozí hodnotu  $p$ . Pro  $m = 1$  byl zvolen vektor hodnot  $p = 0.9$ . Pro  $m \geq 3$  byly komponenty 4, 5, 8, 9 a 10 vyhodnoceny jako nedůležité a jejich indexy byly odebrány z množiny  $I$ , díky tomu bylo zaznamenáno zkrácení výpočetního času metody IS.
2. Ve druhé variantě metody IS bylo použito rovněž  $M = 10^5$  pokusů na iteraci, počet iterací provedených algoritmem 4.26 se s rostoucím  $m$  zvyšuje, viz B.6. Dále bylo pro zpřesnění parametrů provedeno 10 pokusů základní metody CE, přičemž jako počáteční vektor  $\eta^{(0)}$  byl použit výstup z algoritmu 4.26. V obou případech byl navíc použit vyhlazovací parametr  $\alpha = 0.1$ . Dosažená přesnost je při stejném rozsahu

výběrového souboru výrazně nižší než v případě ostatních metod, s rostoucím  $m$  se navíc přesnost snižuje a je zde možné pozorovat degeneraci pravděpodobnosti, která vede ke zkreslení výsledku z důvodu nízké pravděpodobnosti vygenerování některých vzorků, viz 4.21.

3. Pomocí podmíněné metody MC byla dosažena nejvyšší přesnost odhadu z testovaných metod, výhodou je rovněž skutečnost, že se přesnost s rostoucím  $m$  téměř nesnižuje. Algoritmus 3b využívající paralelizaci pomocí CUDA dosahoval přibližně 140krát nižšího výpočetního času oproti sekvenční implementaci 3a.

V tabulce 8.2 jsou uvedeny výsledky pro hodnoty  $m = 2$  a  $m = 7$ , kompletní výsledky testování lze nalézt v tabulkách B.5 až B.9 v příloze.

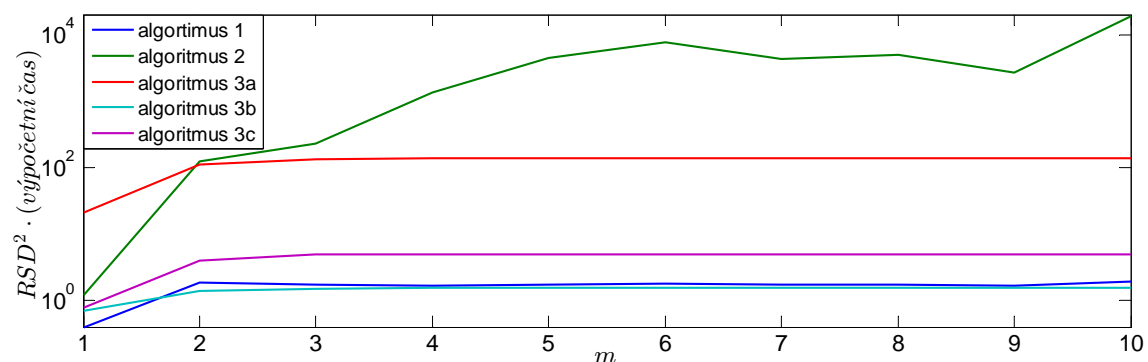
Algoritmus	$m = 2$			$m = 7$		
	$\hat{\ell}$	RSD	výpočetní čas [s]	$\hat{\ell}$	RSD	výpočetní čas [s]
1	$2.01 \cdot 10^{-5}$	0.79%	2.94	$1.98 \cdot 10^{-20}$	0.77%	2.93
2	$2.06 \cdot 10^{-5}$	4.70%	5.62	$3.07 \cdot 10^{-21}$	23.1%	8.16
3a	$1.99 \cdot 10^{-5}$	0.56%	344.6	$1.98 \cdot 10^{-20}$	0.63%	344.3
3b	$1.99 \cdot 10^{-5}$	0.56%	2.45	$2.01 \cdot 10^{-20}$	0.63%	2.44
3c	$2.01 \cdot 10^{-5}$	0.56%	8.47	$2.02 \cdot 10^{-20}$	0.63%	8.41

Tabulka 8.2: Výsledky příkladu 8.6

Graf porovnává pro hodnoty  $m \in \{1, 2, \dots, 10\}$  efektivitu jednotlivých algoritmů. Jako měřítko efektivity byla použita veličina

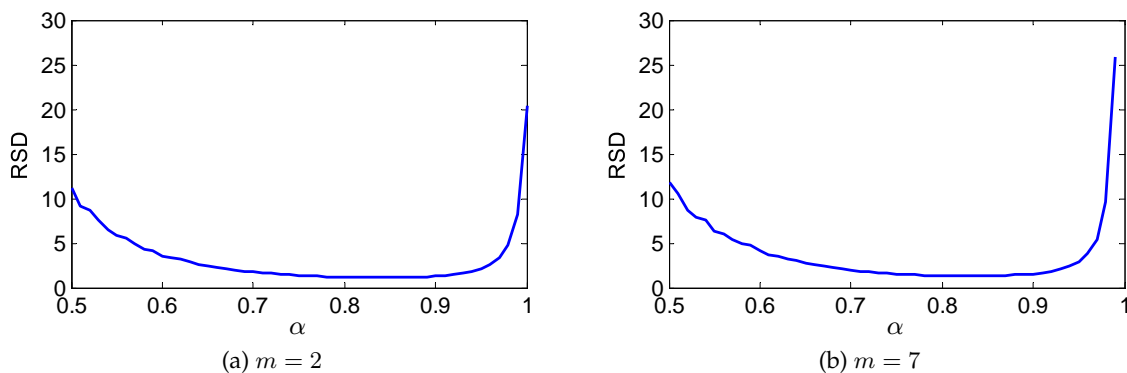
$$RSD^2 \cdot (\text{výpočetní čas}), \quad (8.4)$$

je tak zahrnuta přesnost i čas výpočtu. Jelikož hodnota  $RSD^2$  je nepřímo úměrná rozsahu náhodného výběru  $N$  a výpočetní čas je v případě MC a příbuzných metod přímo úměrný  $N$ , nezávisí hodnota (8.4) na rozsahu náhodného výběru, s rostoucím  $N$  pouze dochází k jejímu zpřesňování.



Obrázek 8.2: Porovnání efektivity jednotlivých algoritmů

Pro  $m = 2$  a  $m = 7$  byla testována rovněž metoda SPB. V obou případech bylo provedeno  $N = 10^6$  pokusů metody IS, jako vektor parametrů IS hustoty byl použit vektor  $\eta = \alpha \cdot p$ , kde  $\alpha \in \langle 0.5, 1 \rangle$ . Závislost RSD na volbě parametru  $\alpha$  ukazují grafy na obrázku 8.3. V případě  $m = 2$  bylo dosaženo minimální hodnoty  $RSD = 1.107\%$  pro  $\alpha = 0.85$  a v případě  $m = 7$  minimální hodnoty  $RSD = 1.276\%$  pro  $\alpha = 0.82$ . Jak ukazuje tabulka 8.2, pomocí výše testovaných algoritmů 1 a 3 byly dosaženy nižší hodnoty RSD.



Obrázek 8.3: Metoda SPB

$m$	$\hat{\ell}$	$s$	$RSD$	čas výpočtu [s]	$RSD^2 \cdot (\text{výpočetní čas})$
1	$1.90 \cdot 10^{-2}$	$1.37 \cdot 10^{-1}$	0.01%	5253	$2.77 \cdot 10^{-1}$
2	$1.99 \cdot 10^{-5}$	$4.50 \cdot 10^{-3}$	0.22%	5144	$2.58 \cdot 10^2$
3	$2.04 \cdot 10^{-8}$	$1.43 \cdot 10^{-4}$	7.00%	5197	$2.55 \cdot 10^5$

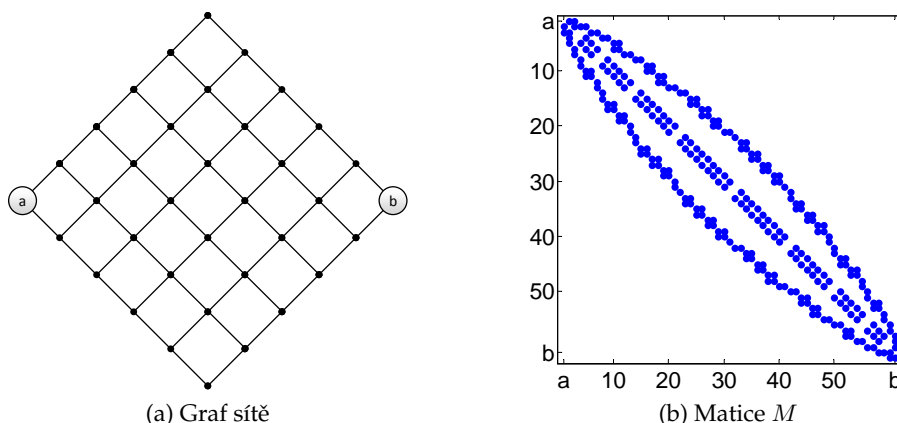
Tabulka 8.3: Výsledky příkladu 8.6 řešeného metodou MC

Pro srovnání byla úloha řešena také pomocí základní metody MC. Výsledky pro  $m \in \{1, 2, 3\}$  a rozsah náhodného výběru  $N = 10^{10}$  ukazuje tabulka 8.3. Pro  $m > 3$  již tento rozsah náhodného výběru nestačil k zaznamenání řídkého jevu.  $\triangle$

Následuje systém, jehož spolehlivost je dána spíše počtem komponent, než bezporuchovostí komponent samotných. Opět se jedná o úlohu typu 8.1.

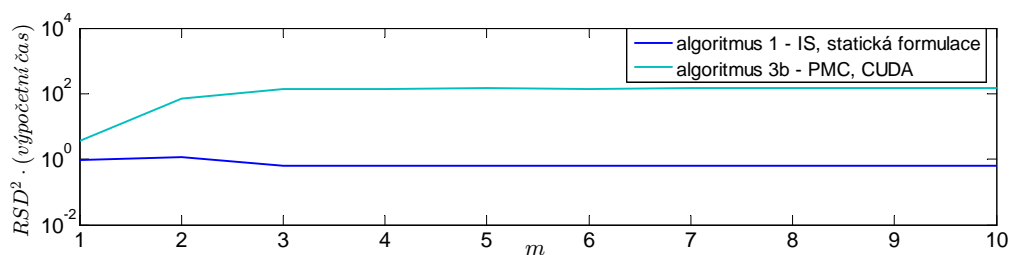
**Příklad 8.7.** Systém je složen ze 60 komponent, strukturu systému a odpovídající matice  $M$  znázorňují obrázky 8.4. Pravděpodobnost provozu jednotlivých komponent je shodná, tj.  $p_i = p$ , kde  $i \in \{1, \dots, 60\}$ , úloha byla opět řešena pro  $p = 1 - 0.1^m$ , kde  $m \in \{1, 2, \dots, 10\}$ .

Pro řešení úlohy byly použity metody 1 (metoda IS aplikovaná na statickou formulaci), 2 (metoda IS aplikovaná na dynamickou formulaci) a 3b (metoda PMC), všechny tyto algoritmy využívají CUDA implementaci. V tabulce B.10 v příloze jsou uvedeny odhady spolehlivosti systému  $\hat{\ell}$  a hodnoty RSD získané pomocí těchto metod pro  $N = 10^6$  náhodných pokusů.



Obrázek 8.4: Znázornění systému z příkladu 8.7

1. V případě algoritmu 1 bylo navíc provedeno 10 iterací metody CE a v každé z nich  $10^5$  náhodných pokusů. Stejně jako v příkladu 8.6 byl pro  $m > 1$  jako počáteční vektor  $\eta^{(0)}$  použit vždy vektor  $\eta^{(10)}$  spočtený pro parametr  $m - 1$ . Tato metoda dosáhla výrazně nižšího rozptylu i výpočetního času než zbylé testované metody. Běh algoritmu pro jednotlivé hodnoty  $m$  trval v průměru 9.2 s.
2. V případě algoritmu 2 bylo pro jednotlivé hodnoty  $m$  provedeno mezi deseti a šestnácti iteracemi metody CE, v každé z nich  $10^5$  náhodných pokusů. Stejně jako při řešení příkladu 8.6 přesnost výsledku se snižující se hodnotou  $\hat{\ell}$  značně klesá, potvrzuje se tedy, že pro odhad pravděpodobnosti poruchy vysoce spolehlivých systémů tento algoritmus není vhodný.
3. Na rozdíl od příkladu 8.6 byla tentokrát pomocí algoritmu 3b dosažena nižší přesnost než pomocí algoritmu 1. Každý běh algoritmu trval přibližně 16.6 s. Algoritmy 3a a 3c již testovány nebyly, neboť dosahují stejné přesnosti jako 3b, pouze v delším čase.

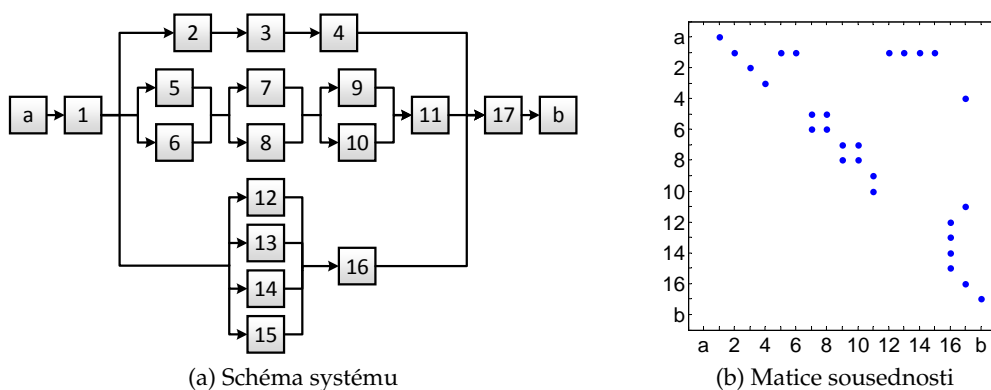


Obrázek 8.5: Porovnání efektivity algoritmů 1 a 3b

Graf na obrázku 8.5 porovnává efektivitu algoritmů 1 a 3b prostřednictvím hodnot  $RSD^2 \cdot (\text{výpočetní čas})$ . Algoritmus 2 do porovnání nebyl zahrnut, neboť pro  $m > 3$  jsou hodnoty RSD z důvodu degenerace LR velmi zkreslené.  $\triangle$

Pro testování algoritmů byl dále použit systém, jehož schéma pochází z [20, str. 122]. Tentokrát se jedná o úlohu typu 8.2

**Příklad 8.8.** Systém je tvořen 17 komponentami, jeho schéma je znázorněno na obrázku 8.6a. Komponentami jsou v tomto případě vrcholy orientovaného grafu, pravděpodobnost provozu  $p_i$  jednotlivých komponent udává tabulka 8.4.



Obrázek 8.6: Znázornění systému z příkladu 8.8

Číslo komponenty	1,17	2,3,4,11,16	5,6,7,8,9,10,12,13,14,15
Pravděpodobnost provozu	0.999	0.99	0.9

Tabulka 8.4: Zadání příkladu 8.8

Pro odhad spolehlivosti systému byly použity algoritmy 1 (metoda IS aplikovaná na statickou formulaci), 2 (metoda IS aplikovaná na dynamickou formulaci) a 3b (metoda PMC) a základní metoda MC, všechny použité implementace využívají technologii CUDA. Výsledky pro rozsah náhodného výběru  $N = 10^6$  jsou uvedeny v tabulce 8.5. V případě metody CE bylo vždy provedeno 10 iterací, v každé  $10^5$  náhodných pokusů.

	$\hat{\ell}$	99% interval spolehlivosti	RSD	výpočetní čas	efektivita
1	$2.01 \cdot 10^{-3}$	$(1.995 \cdot 10^{-3}, 2.026 \cdot 10^{-3})$	0.29%	4.38 s	0.368
2	$2.00 \cdot 10^{-3}$	$(1.978 \cdot 10^{-3}, 2.032 \cdot 10^{-3})$	0.52%	4.91 s	1.327
3a	$2.01 \cdot 10^{-3}$	$(1.972 \cdot 10^{-3}, 2.047 \cdot 10^{-3})$	0.71%	2.43 s	1.225
MC	$2.02 \cdot 10^{-3}$	$(1.906 \cdot 10^{-3}, 2.138 \cdot 10^{-3})$	2.22%	0.69 s	3.401

Tabulka 8.5: Výsledky příkladu 8.8

V tomto případě je efektivita všech tří metod redukce rozptylu srovnatelná. Výsledná hodnota nespolehlivost systému nespadá do kategorie řídkých jevů, opět se tedy ukázalo, že pro vyšší hodnoty  $\hat{\ell}$  lze použít i algoritmus 2.  $\triangle$

### 8.5.1 Zhodnocení

Mezi jednotlivými implementacemi metody PMC dosahuje jednoznačně nejlepších výsledků algoritmus 3b využívající technologii CUDA. Zbývá tak porovnat algoritmy 1, 2 a 3b z hlediska možností jejich využití.

Zaměříme se nejprve na algoritmus 2, tedy metodu IS aplikovanou na dynamickou formulaci systému, která využívá exponenciální rozdělení, jehož parametry jsou odhadovány pomocí metody CE pro kvantifikaci řídkých jevů. Testování ukázalo, že tato metoda není vhodná v případě vysoce spolehlivých systémů, kde hodnota  $\ell$  je velmi malá, např. menší než  $10^{-6}$ . Pro odhad pravděpodobností okolo  $\ell = 10^{-2}$  je tato metoda vhodná, její efektivita je srovnatelná s ostatními metodami.

Algoritmy 1 (metoda IS aplikovaná na statickou formulaci) a 3b (metoda PMC) dosahovaly výborných výsledků ve všech testovaných případech, jsou tedy vhodné i pro odhad spolehlivosti vysoce spolehlivých systémů. V příkladu 8.7 uvažujícím systém o vysokém počtu komponent se úspěšně uplatnil algoritmus 8.3 využívající *screening*, díky němuž byly do LR byly zahrnuty pouze významné komponenty. Užitečnou modifikací bylo rovněž používání vhodného počátečního vektoru  $\eta^{(0)}$ . V tomto případě byla řešena vždy sada úloh se zvyšující se hodnotou  $m$  a tím i spolehlivostí systému, proto se nabízelo použití vektoru  $\eta^{(k_{\max})}$  získaného pro nižší parametr  $m$ . Tento postup se však dá využít i obecně při odhadu velmi nízkých hodnot  $\ell$ , viz. algoritmus 8.9.

---

**Algoritmus 8.9** (Modifikace algoritmu 8.3 pro kvantifikaci řídkých jevů). Vstupem algoritmu je vektor  $\mathbf{p}$  počátečních parametrů Bernoulliho rozdělení.

1. Zvolit vhodnou kladnou rostoucí posloupnost  $(\alpha_k)_{k=1}^m$  o délce  $m$ , pro kterou platí  $\alpha_m = 1$ . Nastavit  $k = 1$ ,  $\eta^{(0)} = \alpha_1 \cdot \mathbf{p}$ .
  2. Určit vektor  $\eta^{(k)}$  jako výstup algoritmu 4.23 základní metody CE se vstupem  $\theta = \alpha_k \cdot \mathbf{p}$  a počátečním vektorem parametrů IS hustoty  $\eta^{(k-1)}$ .
  3. Je-li  $k < m$ , navýšit proměnnou  $k$  o 1 a pokračovat krokem 2.
  4. Použít  $\eta^{(m)}$  jako počáteční vektor parametrů metody CE pro kvantifikaci spolehlivosti sítě, viz algoritmus 8.3.
- 

## 8.6 Analýza citlivosti

S odhadem spolehlivosti systémů s nezávislými prvky souvisí rovněž analýza citlivosti, která slouží k vyhodnocení vlivu jednotlivých komponent na spolehlivost systému jako celku. Na základě výsledků analýzy citlivosti může být reálný systém zdokonalen nahrazením důležitých komponent spolehlivějšími prvky. Analýzu citlivosti lze využít i opačným způsobem a nahradit komponenty s nízkým vlivem na spolehlivost systému komponentami méně kvalitními, což povede ke snížení provozních nákladů, zatímco spolehlivost systému zůstane téměř stejná.

Nespolehlivost  $\ell$  je zde chápána jako funkce parametrů pravděpodobnostní funkce, tedy  $\ell = \ell(\mathbf{p})$ , kde  $\mathbf{p}$  je vektorem parametrů Bernoulliho rozdělení. Pro odhad parciálních derivací funkce  $\ell(\mathbf{p})$  podle jednotlivých parametrů rozdělení pravděpodobnosti lze použít metodu SF, která poskytuje jejich nestranný odhad. Nejprve je tedy třeba určit funkci  $\mathcal{S}(\mathbf{p}; \mathbf{b})$ . Pravděpodobnostní funkcí náhodného vektoru  $\mathbf{B}$  je funkce

$$f(\mathbf{b}; \mathbf{p}) = \prod_{k=1}^n p_k^{b_k} (1 - p_k)^{1-b_k}.$$

Pro parciální derivace logaritmu pravděpodobnostní funkce platí

$$\frac{\partial}{\partial p_i} \ln f(\mathbf{b}; \mathbf{p}) = \frac{\partial}{\partial p_i} \sum_{k=1}^n \ln \left( p_k^{b_k} (1 - p_k)^{1-b_k} \right) = \begin{cases} \frac{1}{p_i - 1} & \text{pro } b_i = 0 \\ \frac{1}{p_i} & \text{pro } b_i = 1, \end{cases}$$

což lze zapsat jako

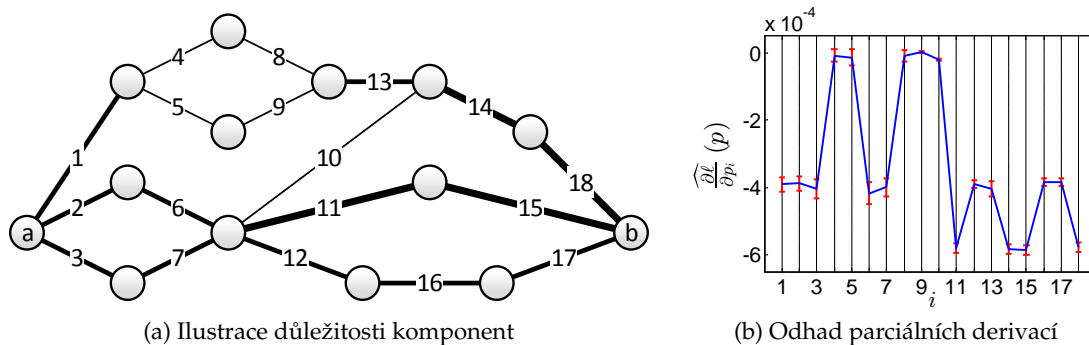
$$\frac{\partial}{\partial p_i} \ln f(\mathbf{b}; \mathbf{p}) = \frac{b_i - p_i}{p_i \cdot (1 - p_i)}.$$

Platí tedy

$$\mathcal{S}(\mathbf{p}; \mathbf{b}) = \left( \frac{b_1 - p_1}{p_1 \cdot (1 - p_1)}, \dots, \frac{b_n - p_n}{p_n \cdot (1 - p_n)} \right).$$

Na výše řešené testovací systémy byla aplikována kombinace metody SF a metody IS, pro odhad parametrů IS hustoty byl použit algoritmus 8.3, tedy metoda CE upravená pro řešení spolehlivosti systémů. V případě všech úloh bylo vykonáno 10 iterací metody CE, v každé iteraci  $M = 10^5$  náhodných pokusů, dále byl použit algoritmus 5.6 pro rozsah náhodného výběru  $N = 10^6$ .

**Příklad 8.10.** Nejprve byla provedena analýza citlivosti systému z příkladu 8.6 pro  $m = 2$ . Výsledné odhady parciálních derivací včetně 99% intervalů spolehlivosti ukazuje graf 8.7b. Obrázek 8.7a slouží k ilustraci důležitosti jednotlivých komponent.

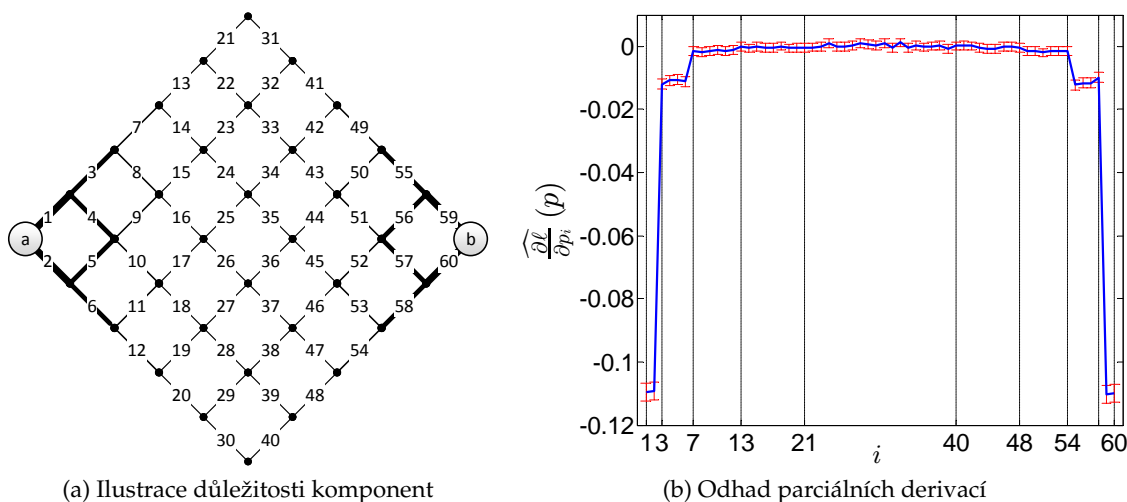


Obrázek 8.7: Analýza citlivosti systému z příkladu 8.6

Jelikož jsou si jednotlivé složky vektoru  $\mathbf{p}$  rovny, lze důležitost komponent porovnat přímo na základě absolutních hodnot parciálních derivací. Za komponenty s největším vlivem na spolehlivost systémů lze tedy považovat komponenty 11, 14, 15 a 18.  $\triangle$



**Příklad 8.11.** V případě systému z příkladu 8.7 lze očekávat, že největší vliv na spolehlivost systému mají komponenty, které přímo sousedí s vrcholy  $a$  a  $b$ . Aby byl systém v provozu, je nutné, aby byla v provozu alespoň jedna z komponent 1 a 2 a alespoň jedna z komponent 59 a 60.



Obrázek 8.8: Analýza citlivosti systému z příkladu 8.7

Výsledky analýzy odpovídají očekávání, s rostoucí vzdáleností od koncových vrcholů se důležitost komponent snižuje. V praxi by tedy bylo vhodné použít spolehlivější komponenty v blízkosti koncových prvků a naopak. Graf odhadů parciálních derivací funkce  $\ell(\mathbf{p})$  pro  $m = 1$  včetně 99% intervalů spolehlivosti znázorňuje obrázek 8.8b.  $\triangle$

**Příklad 8.12.** Ze struktury systému z příkladu 8.8 plyne, že pro provozuschopnost systému je nutné, aby byly v provozu komponenty 1 a 17. Podle očekávání je absolutní hodnota odhadu parciálních derivací funkce  $\ell(\mathbf{p})$  podle  $p_1$  a  $p_{17}$  nejvyšší a ostatní parciální derivace jsou blízké nule.

V tomto případě se jednotlivé složky vektoru  $\mathbf{p}$  liší, důležitost jednotlivých komponent by tedy měla být stanovena na základě hodnot

$$\left( \left| p_i \cdot \left( \widehat{\nabla} \ell(\mathbf{p}) \right)_i \right| \right)_{i=1, \dots, 17},$$

jak bylo ilustrováno v příkladu 5.3.

Pro původní vektor  $\mathbf{p}$  byl získán odhad spolehlivosti  $\widehat{\ell} = 2.01 \cdot 10^{-3}$ . Nahrazením komponent, které jsou v provozu s pravděpodobností  $p_i = 0.9$  komponentami se spolehlivostí  $\tilde{p}_i = 0.75$ , mohou být sníženy náklady na provoz systému, jeho spolehlivost se však změní pouze minimálně, s novými hodnotami vychází odhad spolehlivosti systému na  $\widehat{\ell}_2 = 2.06 \cdot 10^{-3}$ .  $\triangle$

## 9 Závěr

Hlavním úkolem této práce bylo provést efektivní stochastickou simulaci DE systémů s důrazem na kvantifikaci řídkých jevů. Tento cíl byl naplněn použitím metod redukce rozptylu založených na metodě Monte Carlo a dále efektivní implementací zahrnující paralelizaci pomocí technologie CUDA.

Nejprve bylo třeba uvést základní teoretické poznatky z oblasti simulace statických a dynamických DE systémů a představit metodu Monte Carlo jako nástroj k vyhodnocení simulačních dat.

Pro redukci rozptylu bylo zvoleno několik metod, které pracují s nestranným odhadem hledané hodnoty. Jedná se o metodu antitetických náhodných veličin, metodu řídicích veličin, podmíněnou metodu Monte Carlo a metodu *importance sampling*. V případě jednotlivých metod byl vždy určen rozptyl příslušného estimátoru, uvedeny předpoklady vedoucí k redukci rozptylu, odvozena chyba odhadu a formulován obecný algoritmus metody. Na základě provedených testů lze obecně zhodnotit jednotlivé metody z hlediska jejich využití. Metoda antitetických veličin je nenáročnou modifikací základní Monte Carlo simulace, přesto však může vést k významné redukci rozptylu. Oproti tomu efektivní použití metody řídicích veličin a podmíněné metody Monte Carlo vyžaduje dobrou znalost simulovaného systému. Metodu IS lze často použít bez znalosti systému, neboť existují mechanismy pro simulační odhad parametrů IS rozdělení, nicméně nevhodná volba parametrů může vést ke zvýšení rozptylu odhadu.

Pro analýzu citlivosti byla studována metoda centrálních diferencí a metoda *score function*. Bylo zdůvodněno, proč o míře vlivu jednotlivých komponent na hledanou hodnotu není vhodné rozhodovat přímo na základě porovnání odhadovaných složek gradientu, a bylo ukázáno, které hodnoty se pro posouzení vlivu komponent hodí lépe.

Práce zahrnuje rovněž implementaci všech zmíněných metod redukce rozptylu a analýzy citlivosti. Metody byly implementovány formou funkcí v jazyce Matlab, které slouží jako univerzální schémata, jež mohou být použita při simulaci statických systémů a dynamických systémů v konečném časovém horizontu. Jednotlivé řešené úlohy se liší zejména výkonností funkcí, která je vstupem zmíněných schémat, efektivita její implementace udává efektivitu celé simulace. Významnou část práce proto tvoří využití paralelních výpočtů na GPU pomocí technologie CUDA.

Představené metody redukce rozptylu byly aplikovány na řešení konkrétního problému dynamické spolehlivosti (nádrž s čerpadly), který se vyskytuje v mnoha odborných článcích a slouží k testování různých simulačních metod. Pro řešení tohoto problému byla nejprve použita základní metoda Monte Carlo, která byla zefektivněna pomocí CUDA. Využití paralelních výpočtů na GPU vedlo k více než stonásobnému snížení výpočetního času oproti ekvivalentní sekvenční implementaci v jazyce Matlab. Pomocí vhodných simulačních metod bylo dále dosaženo redukce rozptylu v řádu desítek až stovek v závislosti na použité metodě a odhadované kvalitě systému. V případě každé z odhadovaných kvalit systému dosáhla nejlepších výsledků jiná z metod. Toto ukazuje, že se nestačí zaměřit pouze na jednu z metod, neboť každou je vhodné použít na jiný typ problému.

Dále byl pomocí studovaných metod řešen obecný problém kvantifikace spolehlivosti systémů s nezávislými prvky. Pro řešení tohoto problému bylo použito více variant metody IS a metoda Permutation Monte Carlo (PMC), která je speciální verzí podmíněné metody Monte Carlo. Všechny metody byly implementovány s využitím paralelních výpočtů, například výpočetní čas metody PMC byl pomocí CUDA a odvozeného explicitního vzorce snížen přibližně 140krát oproti původní sekvenční implementaci. Jelikož se jednalo o jiný typ formulace, než který řeší klasická metoda CE určená pro kvantifikaci řídkých jevů, byl navrhnut algoritmus, který umožňuje použití metody CE i v tomto případě. Pro zdokonalení tohoto algoritmu byla dále využita myšlenka *screening* metody, která se používá pro zabránění degenerace LR. Výsledný algoritmus dosahuje srovnatelných výsledků jako metoda PMC, v případě systémů s nepravidelnou strukturou a vysokým počtem komponent dosahuje dokonce výsledků výrazně lepších z hlediska výpočetního času i dosažené redukce rozptylu. Se zvyšující se spolehlivostí systémů se zvyšovala rovněž dosažená redukce rozptylu, například v případě testovacího systému s pravděpodobností poruchy  $10^{-5}$  se rozptyl snížil přibližně 1600krát a při pravděpodobnosti poruchy  $10^{-8}$  přibližně  $1.3 \cdot 10^6$ krát oproti základní metodě MC.

Za hlavní přínosy této práce lze považovat dosažení výrazného snížení výpočetního času simulací pomocí paralelních výpočtů na GPU a navržení algoritmů, které jsou vhodné pro kvantifikaci řídkých jevů, jež nastávají například při odhadu spolehlivosti systémů s nezávislými prvky.

Téma práce je široké a nabízí tak mnoho prostoru pro další pokračování. Studované metody lze například aplikovat při analýze systémů hromadné obsluhy a v řadě dalších odvětví. Možným pokračováním práce je rovněž použití metody Multilevel Monte Carlo, která je založena na metodě řídicích veličin, při řešení stochastických diferenciálních rovnic.

## 10 Literatura

- [1] RUBINSTEIN, Reuven Y.; KROESE, Dirk P. *Simulation and the Monte Carlo method*. John Wiley & Sons, Inc., 2008. ISBN 978-0-470-17794-5.
- [2] KROESE, Dirk P.; TAIMRE, Thomas; BOTEV, Zdravko I. *Handbook of Monte Carlo Methods*. John Wiley & Sons, Inc., 2011. ISBN 978-0-470-17793-8
- [3] LITCHMANNOVÁ, Martina. *Vybrané kapitoly z pravděpodobnosti* [online]. VŠB-TU Ostrava, Fakulta elektrotechniky a informatiky, 2011. Dostupné z <mi21.vsb.cz> [cit. 4. března 2015].
- [4] *MATLAB Documentation* [online]. The MathWorks, Inc., 2015. Dostupné z <http://www.mathworks.com/help/matlab/> [cit. 20. dubna 2015].
- [5] DUBI, A. *Monte Carlo Applications in Systems Engineering*. John Wiley & Sons, Inc., 1999. ISBN 0-471-98172-9.
- [6] LITCHMANNOVÁ, Martina. *Úvod do statistiky* [online]. VŠB-TU Ostrava, Fakulta elektrotechniky a informatiky, 2011. Dostupné z <mi21.vsb.cz> [cit. 4. března 2015].
- [7] Kleijnen, Jack P.C.; Ridder, Ad A. N.; Rubinstein, Reuven Y. *Variance Reduction Techniques in Monte Carlo Methods*. 2010.
- [8] RUBINSTEIN, Reuven Y.; KROESE, Dirk P. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2004. ISBN 978-0-387-21240-1.
- [9] BOUCHALA, Jiří. *Variační metody* [online]. Ostrava, 2012. Dostupné z <mi21.vsb.cz> [cit. 11. dubna 2015].
- [10] *CUDA C Programming Guide*. NVIDIA Corporation, 2015. PG-02829-001\_v7.0.
- [11] MARSEGUERRA, M.; ZIO, E. Monte Carlo approach to PSA for dynamic process systems. *Reliability Engineering & System Safety*. 1996. 52.3: 227-241.
- [12] ŠKŇOUŘILOVÁ, Petra. *Dynamické spolehlivostní modely na bázi Petriho sítí*. Ostrava, 2009. Dizertační práce. VŠB-TU Ostrava.
- [13] CODETTA-RAITERI, D.; BOBBIO, A. Solving dynamic reliability problems by means of ordinary and Fluid Stochastic Petri Nets. *Advances in Safety and Reliability*. Taylor & Francis Group, London, 2005. ISBN 0-415-38340-4.
- [14] BROÏ, Perrine; CHRAIBI, Hassane; DONAT, Roland. Using Dynamic Bayesian Networks to solve a dynamic reliability problem. *Advances in Safety, Reliability and Risk Management*. Taylor & Francis Group, London, 2012. ISBN 978-0-415-68379-1.
- [15] BRIŠ, Radim. *Inovační metody pro ocenění spolehlivosti proků a systémů*. 1. vyd. Ostrava: VŠB-TU Ostrava, 2007. ISBN 978-80-248-1596-1.

- 
- [16] KUBESA, Michael. *Základy diskrétní matematiky* [online]. VŠB-TU Ostrava, 2011. Dostupné z <mi21.vsb.cz> [cit. 24. března 2015].
- [17] KOVÁŘ, Petr. *Teorie grafů* [online]. Ostrava, 2012. Dostupné z <mi21.vsb.cz> [cit. 24. března 2015].
- [18] Ross, Sheldon M. *Introduction to Probability Models*. Ninth Edition. Elsevier Inc., 2007. ISBN 978-0-12-598062-3.
- [19] BRIŠ, Radim, DOMESOVÁ, Simona. New Computing Technology in Reliability Engineering. *Mathematical Problems in Engineering*, 2014. Article ID 187362. ISSN 1563-5147.
- [20] RAUSAND, Marvin, HØYLAND, Arnljot. *System reliability theory*. 2<sup>nd</sup> edition. Wiley, 2004. ISBN 0-471-47133-X.

## A Přehled používaných rozdělení pravděpodobnosti

Bylo převzato značení používané v [1]. Pro každé rozdělení je uvedeno značení faktu, že náhodná veličina  $X$  pochází z daného rozdělení, a hustota pravděpodobnosti. Pro generování z těchto rozdělení jsou používány vestavěné funkce jazyka Matlab a inverzní transformace.

### A.1 Rovnoměrné rozdělení

Značení	Hustota pravděpodobnosti	Poznámky
$X \sim \mathcal{U}(\alpha, \beta)$	$f(x; \alpha, \beta) = \frac{1}{\beta - \alpha}$	$x \in \langle \alpha, \beta \rangle, \alpha < \beta$

### A.2 Bernoulliho rozdělení

Značení	Hustota pravděpodobnosti	Poznámky
$X \sim \text{Ber}(p)$	$f(x; p) = p^x (1 - p)^{1-x}$	$x \in \{0, 1\}, 0 \leq p \leq 1$

### A.3 Exponenciální rozdělení

Značení	Hustota pravděpodobnosti	Poznámky
$X \sim \text{Exp}(\lambda)$	$f(x; \lambda) = \lambda \exp(-\lambda x)$	$x \in \mathbb{R}^+, \lambda > 0$

V případě exponenciálního rozdělení je častá rovněž parametrizace pomocí střední hodnoty  $\mu = \lambda^{-1}$ , viz následující tabulka.

Značení	Hustota pravděpodobnosti	Poznámky
$X \sim \text{Exp}(\mu)$	$f(x; \lambda) = \frac{1}{\mu} \exp\left(-\frac{x}{\mu}\right)$	$x \in \mathbb{R}^+, \mu > 0$

### A.4 Weibullovo rozdělení

Značení	Hustota pravděpodobnosti	Poznámky
$X \sim \text{Weib}(\alpha, \lambda)$	$f(x; \alpha, \lambda) = \alpha \lambda (\lambda x)^{\alpha-1} \exp(-(\lambda x)^\alpha)$	$x \in \mathbb{R}^+, \alpha > 0, \lambda > 0$

### A.5 Normální rozdělení

Značení	Hustota pravděpodobnosti	Poznámky
$X \sim \mathcal{N}(\mu, \sigma^2)$	$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$	$x \in \mathbb{R}, \sigma > 0, \mu \in \mathbb{R}$

## B Výsledky testování a vybrané zdrojové kódy

### B.1 Výsledky analýzy citlivosti

V následujících tabulkách jsou uvedeny výsledky analýzy citlivosti systému z příkladu 3.4. Jedná se o odhad gradientu funkcí  $\ell(\boldsymbol{\theta})$  a  $\ell(\boldsymbol{\mu})$  pomocí metody centrálních diferencí, viz příklad 5.3, a pomocí metody SF, viz příklad 5.7. Následující intervaly spolehlivosti byly určeny pro koeficient spolehlivosti  $\alpha = 0.05$ . Hodnoty  $N_\varepsilon$  značí počet náhodných pokusů nutných pro dosažení přesnosti  $\varepsilon = 0.1$  s pravděpodobností 0.95.

$i$	$\left(\widehat{\nabla\ell}(\boldsymbol{\theta})\right)_i$	$s$	$s^2$	$ RSD $	interval spolehlivosti	$N_\varepsilon$
1	-0.185	0.682	0.465	0.37%	$(-0.18612, -0.18345)$	179
2	-1.012	3.354	11.251	0.33%	$(-1.0181, -1.0049)$	4322
3	-7.26	9.433	88.979	0.13%	$(-7.2789, -7.2419)$	$3.4 \cdot 10^4$
4	-11.324	17.191	295.519	0.15%	$(-11.3572, -11.2899)$	$1.1 \cdot 10^5$
5	-10.288	25.511	650.831	0.25%	$(-10.3376, -10.2376)$	$2.5 \cdot 10^5$
6	-28.929	38.508	1482.827	0.13%	$(-29.0052, -28.8543)$	$5.7 \cdot 10^5$
7	-21.883	51.982	2702.175	0.24%	$(-21.9845, -21.7808)$	$1.0 \cdot 10^6$

Tabulka B.1: Odhad  $\nabla\ell(\boldsymbol{\theta})$  pomocí metody centrálních diferencí

$i$	$\left(\widehat{\nabla\ell_2}(\boldsymbol{\mu})\right)_i$	$s$	$s^2$	$ RSD $	interval spolehlivosti	$N_\varepsilon$
1	0.187	0.685	0.470	0.37%	$(0.18535, 0.18804)$	180
2	0.253	0.840	0.705	0.33%	$(0.25114, 0.25443)$	271
3	0.809	1.050	1.103	0.13%	$(0.80677, 0.81089)$	424
4	0.704	1.072	1.150	0.15%	$(0.7022, 0.70641)$	442
5	0.410	1.022	1.045	0.25%	$(0.40828, 0.41229)$	401
6	0.804	1.074	1.154	0.13%	$(0.80202, 0.80623)$	443
7	0.446	1.065	1.135	0.24%	$(0.44415, 0.44833)$	436

Tabulka B.2: Odhad  $\nabla\ell_2(\boldsymbol{\mu})$  pomocí metody centrálních diferencí

$i$	$(\widehat{\nabla \ell}(\theta))_i$	$s$	$s^2$	$ RSD $	interval spolehlivosti	$N_\varepsilon$
1	-0.197	18.231	332.364	9.25%	(-0.23283, -0.16137)	$1.3 \cdot 10^5$
2	-1.001	38.168	1456.813	3.81%	(-1.0753, -0.92572)	$5.6 \cdot 10^5$
3	-7.289	69.274	4798.907	0.95%	(-7.4248, -7.1533)	$1.8 \cdot 10^6$
4	-11.404	98.634	9728.562	0.86%	(-11.5977, -11.211)	$3.7 \cdot 10^6$
5	-10.195	120.452	$1.45 \cdot 10^4$	1.18%	(-10.4312, -9.959)	$5.6 \cdot 10^6$
6	-28.809	175.533	$3.08 \cdot 10^4$	0.61%	(-29.1533, -28.4652)	$1.2 \cdot 10^7$
7	-22.417	197.939	$3.9 \cdot 10^4$	0.88%	(-22.8046, -22.0287)	$1.5 \cdot 10^7$

Tabulka B.3: Odhad  $\nabla \ell(\theta)$  pomocí metody SF

$i$	$(\widehat{\nabla \ell_2}(\mu))_i$	$s$	$s^2$	$ RSD $	interval spolehlivosti	$N_\varepsilon$
1	0.174	18.158	329.727	10.44%	(0.13841, 0.20959)	$1.3 \cdot 10^5$
2	0.252	9.557	91.327	3.79%	(0.23359, 0.27105)	$3.5 \cdot 10^4$
3	0.808	7.698	59.257	0.95%	(0.79294, 0.82312)	$2.3 \cdot 10^4$
4	0.713	6.174	38.118	0.87%	(0.70056, 0.72476)	$1.5 \cdot 10^4$
5	0.414	4.821	23.244	1.16%	(0.40482, 0.42372)	8929
6	0.809	4.929	24.297	0.61%	(0.79963, 0.81895)	9333
7	0.447	4.022	16.175	0.90%	(0.4387, 0.45446)	6214

Tabulka B.4: Odhad  $\nabla \ell_2(\mu)$  pomocí metody SF

## B.2 Vybrané zdrojové kódy

Jedná se o CUDA kódy funkcí  $G(\pi)$  a  $H(b)$  používané při kvantifikaci spolehlivosti systémů.

```

1  __global__ void CUDA_G(int n, unsigned short int *permutation, float *v,
2    unsigned short int *crit, float *nu)
3  {
4    // n          pocet komponent
5    // permutation permutace komponent
6    // v          vystupni vektor
7    // crit       kriticka hodnota
8    // nu         vektor nu predpocteny na CPU
9    int idx = blockIdx.x * blockDim.x + threadIdx.x; // cislo vlakna
10   int b=crit[idx];
11   float res=0;
12   float w;
13   for (int i=b-1;i>=0;i--)
14   {
15     w=1;
16     for(int j=0;j<b;j++)
17     {

```



```

17         if(j!=i)
18         {
19             w=w*nu[idx*n+j]/(nu[idx*n+j]-nu[idx*n+i]);
20         }
21     }
22     res=res+w*exp(-nu[idx*n+i]);
23 }
24 v[idx]=res;
25 }

```

Výpis B.3: CUDA kernel funkce  $G(\pi)$ 

```

1  __global__ void H_CUDA(int n, float *B, int *v, unsigned int *M)
2  {
3      /// Urci, zda je system pro danou konfiguraci komponent v provozu.
4      // n                pocet komponent
5      // B                konfigurace komponent
6      // v                vystupni vektor
7      // M                matice sousednosti prevedena na vektor uint32
8      int idx = blockIdx.x * blockDim.x + threadIdx.x;    // cislo vlakna
9      const int CH=32;
10     int range = (n+2-1)/CH+1;    // kolik int32 je potreba
11     unsigned int M1[16]={0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
12     unsigned int M2[16];
13     unsigned int M3[16]={0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
14     int working=0;    // hodnota 1 znaci pruchodnost systemu
15     int M2_empty;    // hodnota 1 znaci navstiveni vseh pristupnych hran
16     for(int i=1; i<n+1; i++)    // nefunkcni hrany zapsany do M3
17     {
18         if(B[idx*n+i-1]==0)
19             M3[i/CH]|=1<<(i%CH);
20     }
21     for(int i=0; i<range; i++)    // prvky, které nejsou v M3, a zároveň
22     {    // sousedi s vrcholem A zapsany do M2
23         M2[i]=(M[i]^M3[i])&M[i];    }
24     while(working==0)
25     {
26         if ( ( M2[range-1]&(1<<((n+1)%CH)) )>0 )
27         {    v[idx]=0;    // pokud M2 obsahuje B, system je v provozu
28             return; }
29         M2_empty=1;
30         for(int i=0; i<range; i++)
31         {    if(M2[i]>0)
32             M2_empty=0; }
33         if (M2_empty)
34         {
35             v[idx]=1;    // pokud je M2 prazdna, system neni v provozu
36             return;
37         }
38         for(int i=1; i<n+1; i++)
39         {    // hrany spojené s M2 zapsany do M1
40             if((M2[i/CH]&(1<<(i%CH)))>0)
41             {    for(int j=0; j<range; j++)
42                 M1[j]=M1[j]|M[range*i+j]; }

```

```

43     for(int i=0; i<range; i++)
44     {
45         M3[i]=M3[i]|M2[i]; // prvky M2 presunuty do M3
46         M2[i]=(M1[i]^M3[i])&M1[i]; // prvky M1, které nejsou v M3
47         M1[i]=0;           } // prvky M1 smazany
48     }
49 }

```

Výpis B.4: CUDA kernel funkce  $H(b)$ 

### B.3 Test algoritmů používaných v sekci 8.5

Tabulky B.5 až B.9 porovnávají jednotlivé algoritmy použité k řešení příkladu 8.6.

$m$	$\hat{\ell}$	$s$	RSD	čas výpočtu [s]			nedůležité komponenty
				CE	IS	celkem	
1	$1.90 \cdot 10^{-2}$	$6.68 \cdot 10^{-2}$	0.36%	1.62	1.52	3.14	-
2	$2.01 \cdot 10^{-5}$	$1.59 \cdot 10^{-4}$	0.79%	1.50	1.44	2.94	4, 5, 8, 9
3	$2.01 \cdot 10^{-8}$	$1.55 \cdot 10^{-7}$	0.77%	1.50	1.43	2.93	4, 5, 8, 9, 10
4	$2.01 \cdot 10^{-11}$	$1.52 \cdot 10^{-10}$	0.76%	1.50	1.43	2.93	4, 5, 8, 9, 10
5	$1.98 \cdot 10^{-14}$	$1.51 \cdot 10^{-13}$	0.76%	1.49	1.43	2.93	4, 5, 8, 9, 10
6	$1.99 \cdot 10^{-17}$	$1.56 \cdot 10^{-16}$	0.79%	1.49	1.43	2.93	4, 5, 8, 9, 10
7	$1.98 \cdot 10^{-20}$	$1.52 \cdot 10^{-19}$	0.77%	1.50	1.43	2.93	4, 5, 8, 9, 10
8	$2.03 \cdot 10^{-23}$	$1.55 \cdot 10^{-22}$	0.76%	1.50	1.43	2.93	4, 5, 8, 9, 10
9	$2.00 \cdot 10^{-26}$	$1.51 \cdot 10^{-25}$	0.76%	1.50	1.43	2.93	4, 5, 8, 9, 10
10	$2.00 \cdot 10^{-29}$	$1.61 \cdot 10^{-28}$	0.81%	1.50	1.43	2.93	4, 5, 8, 9, 10

Tabulka B.5: Algoritmus 1 - metoda IS pro statickou formulaci (příklad 8.6)

$m$	$\hat{\ell}$	$s$	RSD	čas výpočtu [s]			počet iterací
				CE	IS	celkem	
1	$1.91 \cdot 10^{-2}$	$9.17 \cdot 10^{-2}$	0.48%	2.68	2.55	5.22	0+10
2	$2.06 \cdot 10^{-5}$	$9.69 \cdot 10^{-4}$	4.70%	3.05	2.57	5.62	2+10
3	$1.21 \cdot 10^{-8}$	$7.13 \cdot 10^{-7}$	5.90%	4.03	2.62	6.65	6+10
4	$1.03 \cdot 10^{-11}$	$1.40 \cdot 10^{-9}$	13.60%	4.81	2.54	7.35	9+10
5	$7.95 \cdot 10^{-15}$	$1.93 \cdot 10^{-12}$	24.29%	5.04	2.54	7.59	10+10
6	$5.37 \cdot 10^{-18}$	$1.68 \cdot 10^{-15}$	31.23%	5.30	2.56	7.86	11+10
7	$3.07 \cdot 10^{-21}$	$7.09 \cdot 10^{-19}$	23.08%	5.58	2.58	8.16	12+10
8	$2.21 \cdot 10^{-24}$	$5.48 \cdot 10^{-22}$	24.75%	5.56	2.54	8.10	12+10
9	$7.35 \cdot 10^{-28}$	$1.34 \cdot 10^{-25}$	18.26%	5.57	2.51	8.08	12+10
10	$1.16 \cdot 10^{-30}$	$5.30 \cdot 10^{-28}$	45.49%	6.61	2.53	9.14	16+10

Tabulka B.6: Algoritmus 2 - metoda IS pro dynamickou formulaci (příklad 8.6)

$m$	$\widehat{\ell}$	$s$	RSD	čas výpočtu [s]
1	$1.90 \cdot 10^{-2}$	$4.68 \cdot 10^{-2}$	0.25%	344.3
2	$1.99 \cdot 10^{-5}$	$1.12 \cdot 10^{-4}$	0.56%	344.6
3	$2.01 \cdot 10^{-8}$	$1.25 \cdot 10^{-7}$	0.62%	345.0
4	$2.01 \cdot 10^{-11}$	$1.26 \cdot 10^{-10}$	0.63%	344.9
5	$1.99 \cdot 10^{-14}$	$1.26 \cdot 10^{-13}$	0.63%	344.6
6	$2.00 \cdot 10^{-17}$	$1.26 \cdot 10^{-16}$	0.63%	345.4
7	$1.98 \cdot 10^{-20}$	$1.26 \cdot 10^{-19}$	0.63%	344.3
8	$2.01 \cdot 10^{-23}$	$1.27 \cdot 10^{-22}$	0.63%	345.0
9	$2.01 \cdot 10^{-26}$	$1.27 \cdot 10^{-25}$	0.63%	345.2
10	$1.98 \cdot 10^{-29}$	$1.25 \cdot 10^{-28}$	0.63%	345.3

Tabulka B.7: Algoritmus 3a - sekvenční implementace metody PMC (příklad 8.6)

$m$	$\widehat{\ell}$	$s$	RSD	čas výpočtu [s]
1	$1.92 \cdot 10^{-2}$	$4.72 \cdot 10^{-2}$	0.27%	2.55
2	$1.99 \cdot 10^{-5}$	$1.12 \cdot 10^{-4}$	0.56%	2.45
3	$1.99 \cdot 10^{-8}$	$1.25 \cdot 10^{-7}$	0.62%	2.46
4	$1.99 \cdot 10^{-11}$	$1.26 \cdot 10^{-10}$	0.63%	2.47
5	$1.97 \cdot 10^{-14}$	$1.25 \cdot 10^{-13}$	0.64%	2.44
6	$1.99 \cdot 10^{-17}$	$1.26 \cdot 10^{-16}$	0.63%	2.45
7	$2.01 \cdot 10^{-20}$	$1.26 \cdot 10^{-19}$	0.63%	2.44
8	$1.98 \cdot 10^{-23}$	$1.26 \cdot 10^{-22}$	0.63%	2.45
9	$1.99 \cdot 10^{-26}$	$1.26 \cdot 10^{-25}$	0.63%	2.46
10	$2.02 \cdot 10^{-29}$	$1.27 \cdot 10^{-28}$	0.63%	2.44

Tabulka B.8: Algoritmus 3b - CUDA implementace metody PMC (příklad 8.6)

$m$	$\widehat{\ell}$	$s$	RSD	čas výpočtu [s]
1	$1.90 \cdot 10^{-2}$	$4.69 \cdot 10^{-2}$	0.25%	8.69
2	$2.01 \cdot 10^{-5}$	$1.13 \cdot 10^{-4}$	0.56%	8.47
3	$1.99 \cdot 10^{-8}$	$1.25 \cdot 10^{-7}$	0.62%	8.48
4	$2.00 \cdot 10^{-11}$	$1.26 \cdot 10^{-10}$	0.63%	8.44
5	$1.99 \cdot 10^{-14}$	$1.26 \cdot 10^{-13}$	0.63%	8.44
6	$1.99 \cdot 10^{-17}$	$1.26 \cdot 10^{-16}$	0.63%	8.42
7	$2.02 \cdot 10^{-20}$	$1.27 \cdot 10^{-19}$	0.63%	8.41
8	$2.01 \cdot 10^{-23}$	$1.26 \cdot 10^{-22}$	0.63%	8.43
9	$2.00 \cdot 10^{-26}$	$1.26 \cdot 10^{-25}$	0.63%	8.44
10	$1.99 \cdot 10^{-29}$	$1.27 \cdot 10^{-28}$	0.63%	8.46

Tabulka B.9: Algoritmus 3c - hybridní implementace metody PMC (příklad 8.6)

Následující tabulka porovnává algoritmy použité při řešení příkladu 8.7.

$m$	Algoritmus 1		Algoritmus 2		Algoritmus 3b	
	$\hat{\ell}$	RSD	$\hat{\ell}$	RSD	$\hat{\ell}$	RSD
1	$2.43 \cdot 10^{-2}$	0.31	$2.42 \cdot 10^{-2}$	0.40	$2.42 \cdot 10^{-2}$	0.46
2	$2.03 \cdot 10^{-4}$	0.34	$2.05 \cdot 10^{-4}$	1.16	$2.04 \cdot 10^{-4}$	2.07
3	$2.00 \cdot 10^{-6}$	0.26	$2.00 \cdot 10^{-6}$	2.14	$2.01 \cdot 10^{-6}$	2.85
4	$2.00 \cdot 10^{-8}$	0.26	$9.92 \cdot 10^{-8}$	1.39	$2.04 \cdot 10^{-8}$	2.93
5	$2.01 \cdot 10^{-10}$	0.26	$1.01 \cdot 10^{-10}$	2.19	$2.01 \cdot 10^{-10}$	2.96
6	$2.00 \cdot 10^{-12}$	0.26	$1.04 \cdot 10^{-12}$	2.08	$2.07 \cdot 10^{-12}$	2.92
7	$2.00 \cdot 10^{-14}$	0.26	$9.90 \cdot 10^{-15}$	2.58	$2.05 \cdot 10^{-14}$	2.93
8	$2.00 \cdot 10^{-16}$	0.26	$9.94 \cdot 10^{-17}$	2.94	$1.99 \cdot 10^{-16}$	2.97
9	$1.99 \cdot 10^{-18}$	0.27	$1.00 \cdot 10^{-18}$	3.28	$2.00 \cdot 10^{-18}$	2.97
10	$1.99 \cdot 10^{-20}$	0.26	$1.02 \cdot 10^{-20}$	4.03	$1.99 \cdot 10^{-20}$	2.98

Tabulka B.10: Výsledky příkladu 8.7

## **C Příloha na CD**

Příložené CD obsahuje

- všechny použité zdrojové kódy,
- vytvořená schémata a grafy v samostatných souborech,
- elektronickou verzi této práce.